Tyler Bade

IT 440

Assignment 4

1) 3 table join

Determine the miles each car has driven and which branch they’re from.

SELECT b.Branch\_Name, c.Car\_ID, SUM(i.Milage) AS MilesDriven

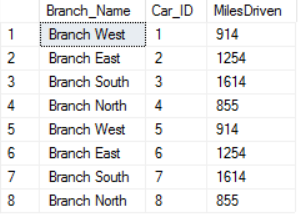
FROM Branch b

INNER JOIN Car c ON b.Branch\_ID = c.Branch\_ID

INNER JOIN Rental r ON b.Branch\_ID = r.Branch\_ID

INNER JOIN Invoice i ON r.Rental\_ID = i.Rental\_ID

GROUP BY b.Branch\_Name, c.Car\_ID;



2) Stored procedure

Get the Miles driven and the customer names of all customers who driven under 100 miles total

CREATE PROC Under100

AS

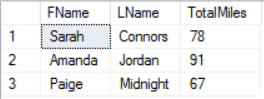
SELECT c.FName, c.LName, SUM(i.Milage) AS TotalMiles

FROM Invoice i

INNER JOIN Customer c ON i.Customer\_ID = c.Customer\_ID

GROUP BY c.FName, c.LName

HAVING SUM(i.Milage) < 100;



3)Cursor

Get the first and last names as well as the IDs of all customers

DECLARE @Customer\_ID AS INT;

DECLARE @FName AS NVARCHAR(20);

DECLARE @LName AS NVARCHAR(20)

DECLARE @CustomerCursor AS CURSOR;

SET @CustomerCursor = CURSOR FOR

SELECT Customer\_ID, FName, LName

FROM Customer;

OPEN @CustomerCursor;

FETCH NEXT FROM @CustomerCursor INTO

@Customer\_ID, @FName, @LName;

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT cast(@Customer\_ID AS NVARCHAR(2)) + ' ' + @FName + ' ' + @LName;

FETCH NEXT FROM @CustomerCursor INTO @Customer\_ID, @FName, @LName;

END

CLOSE @CustomerCursor;

DEALLOCATE @CustomerCursor;



4) CTE

Obtain the names and invoice numbers of all customers that have driven over 100 miles.

;WITH Miles AS(

SELECT Invoice\_ID, Customer\_ID, Milage

FROM Invoice

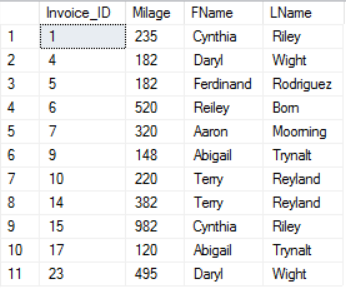
WHERE Milage > 100

)

SELECT m.Invoice\_ID, m.Milage, c.FName, c.LName

FROM Customer c

INNER JOIN Miles m on m.Customer\_ID = c.Customer\_ID;



5) Non-clustered index

Index for the Car table that sorts Price\_Per\_Day in descending order

CREATE NONCLUSTERED INDEX CarDex

ON Car

(Price\_Per\_Day DESC)



6) Join + Index

Find Car ID and Branch Name and Price per day of all SUVs

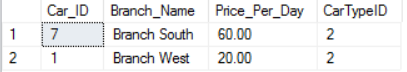
SELECT c.Car\_ID, b.Branch\_Name, c.Price\_Per\_Day, c.CarTypeID

FROM Car c WITH (INDEX(CarDex))

JOIN Branch b ON c.Branch\_ID = b.Branch\_ID

WHERE c.CarTypeID = 2;

GO



7) After trigger

If a car is driven more than 100 miles in a single invoice, a reminder is sent to check the car

CREATE TRIGGER MileCheck ON Invoice

AFTER INSERT

AS

IF EXISTS (SELECT \*

FROM Invoice

WHERE Milage > 100

)

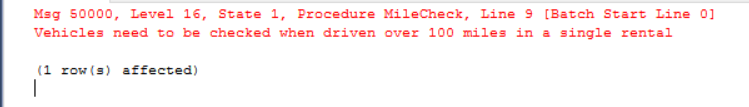
BEGIN

RAISERROR ('Vehicles need to be checked when driven over 100 miles in a single rental', 16, 1);

RETURN

END;

GO



8) View

Car ID and total miles on each car

CREATE VIEW CarMiles AS

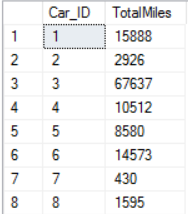
SELECT c.Car\_ID, c.Initial\_Milage + SUM(i.Milage) AS TotalMiles

FROM Car c

LEFT OUTER JOIN Rental r ON c.Car\_ID = r.Car\_ID

INNER JOIN Invoice i ON r.Rental\_ID = i.Rental\_ID

GROUP BY c.Car\_ID, c.Initial\_Milage;



9) Pivot

Total Initial mileage for all cars from each branch

SELECT [1] AS West, [2] AS East, [3] AS North, [4] AS South

FROM

(SELECT Initial\_Milage, Branch\_ID

FROM Car) c

PIVOT

(

SUM(Initial\_Milage)

FOR Branch\_ID IN

([1],[2],[3],[4])

) AS pvt



10) Login

USE [master]

GO

CREATE LOGIN [JohnSmith] WITH PASSWORD = 'default'

GO

CREATE SERVER ROLE [datareader];

ALTER SERVER ROLE [datareader] ADD MEMBER [JohnSmith]

GO

