**PROJECT**

**SQL Query Optimization**

**Database schema:**

Diagram, schematic

Description automatically generated

The healthcare department attempting to use the resources more efficiently. It already has some queries that are being used for different purposes. The management suspects that these queries might not be efficient so they have requested to optimize the existing queries wherever necessary.

Given are some queries written in SQL server which may be optimized if necessary.

**Query 1:**

-- For each age(in years), how many patients have gone for treatment?

SELECT DATEDIFF(hour, dob , GETDATE())/8766 AS age, count(\*) AS numTreatments

FROM Person

JOIN Patient ON Patient.patientID = Person.personID

JOIN Treatment ON Treatment.patientID = Patient.patientID

group by DATEDIFF(hour, dob , GETDATE())/8766

order by numTreatments desc;

SELECT TIMESTAMPDIFF(YEAR, dob , CURDATE()) AS age, count(\*) AS numTreatments

FROM Patient

JOIN Treatment USING(patientid)

group by age

order by numTreatments desc;

-- 1. Grouped by age alias directly

-- 2. Removed join on Person table

-- 3. Replaced ON clause with USING

**Query 2:**

-- For each city, Find the number of registered people, number of pharmacies, and number of insurance companies.

drop table if exists T1;

drop table if exists T2;

drop table if exists T3;

select Address.city, count(Pharmacy.pharmacyID) as numPharmacy

into T1

from Pharmacy right join Address on Pharmacy.addressID = Address.addressID

group by city

order by count(Pharmacy.pharmacyID) desc;

select Address.city, count(InsuranceCompany.companyID) as numInsuranceCompany

into T2

from InsuranceCompany right join Address on InsuranceCompany.addressID = Address.addressID

group by city

order by count(InsuranceCompany.companyID) desc;

select Address.city, count(Person.personID) as numRegisteredPeople

into T3

from Person right join Address on Person.addressID = Address.addressID

group by city

order by count(Person.personID) desc;

select T1.city, T3.numRegisteredPeople, T2.numInsuranceCompany, T1.numPharmacy

from T1, T2, T3

where T1.city = T2.city and T2.city = T3.city

order by numRegisteredPeople desc;

select city, count(personID) as `Person Count`, count(pharmacyID) as `Pharmacy Count`, count(companyID) as `Company Count` from address a

left join insurancecompany i using (addressID)

left join person p using (addressID)

left join pharmacy p2 using (addressID)

group by city

order by `Person Count` desc, `Pharmacy Count` desc, `Company Count` desc;

-- Replaced the entire query with a single select statement with all joins in one.

**Query 3:**

-- Total quantity of medicine for each prescription prescribed by Ally Scripts

-- If the total quantity of medicine is less than 20 tag it as "Low Quantity".

-- If the total quantity of medicine is from 20 to 49 (both numbers including) tag it as "Medium Quantity".

-- If the quantity is more than equal to 50 then tag it as "High quantity".

select

C.prescriptionID, sum(quantity) as totalQuantity,

CASE WHEN sum(quantity) < 20 THEN 'Low Quantity'

WHEN sum(quantity) < 50 THEN 'Medium Quantity'

ELSE 'High Quantity' END AS Tag

FROM Contain C

JOIN Prescription P

on P.prescriptionID = C.prescriptionID

JOIN Pharmacy on Pharmacy.pharmacyID = P.pharmacyID

where Pharmacy.pharmacyName = 'Ally Scripts'

group by C.prescriptionID;

select

prescriptionID, sum(quantity) as totalQuantity,

CASE WHEN sum(quantity) < 20 THEN 'Low Quantity'

WHEN sum(quantity) < 50 THEN 'Medium Quantity'

ELSE 'High Quantity' END AS Tag

FROM Contain

JOIN Prescription using (prescriptionID)

JOIN Pharmacy using (pharmacyID)

where pharmacyName = 'Ally Scripts'

group by prescriptionID;

-- Replaced ON clause with USING

**Query 4:**

-- The total quantity of medicine in a prescription is the sum of the quantity of all the medicines in the prescription.

-- Select the prescriptions for which the total quantity of medicine exceeds

-- the avg of the total quantity of medicines for all the prescriptions.

drop table if exists T1;

select Pharmacy.pharmacyID, Prescription.prescriptionID, sum(quantity) as totalQuantity

into T1

from Pharmacy

join Prescription on Pharmacy.pharmacyID = Prescription.pharmacyID

join Contain on Contain.prescriptionID = Prescription.prescriptionID

join Medicine on Medicine.medicineID = Contain.medicineID

join Treatment on Treatment.treatmentID = Prescription.treatmentID

where YEAR(date) = 2022

group by Pharmacy.pharmacyID, Prescription.prescriptionID

order by Pharmacy.pharmacyID, Prescription.prescriptionID;

select \* from T1

where totalQuantity > (select avg(totalQuantity) from T1);

with T as (

select prescriptionID, sum(quantity) as totalQuantity

from Prescription p join Contain c using (prescriptionID)

group by prescriptionID

order by prescriptionID

)

select \* from T

where totalQuantity > (select avg(totalQuantity) from T);

-- 1. Removed unnecesary tables and joins which were not asked by the problem statement.

-- 2. Replaced ON clause with USING in joins

**Query 5:**

-- Select every disease that has 'p' in its name, and

-- the number of times an insurance claim was made for each of them.

SELECT Disease.diseaseName, COUNT(\*) as numClaims

FROM Disease

JOIN Treatment ON Disease.diseaseID = Treatment.diseaseID

JOIN Claim On Treatment.claimID = Claim.claimID

WHERE diseaseName IN (SELECT diseaseName from Disease where diseaseName LIKE '%p%')

GROUP BY diseaseName;

SELECT diseaseName, COUNT(\*) as numClaims

FROM Disease

JOIN Treatment using (diseaseID)

JOIN Claim using(claimID)

WHERE diseaseName LIKE '%p%'

GROUP BY diseaseName;

-- 1. Removed unnecesary sub-query.

-- 2. Replaced ON clause with USING in joins.