Phase II Project

**Three phases: Create the database with the code below, load the data properly from the Excel file created by the professor, and query data based on the statements below.**

**\*This is what the teacher gave us. It is different from what we created in Phase I\***

CREATE TABLE PhysicianSpecialties

(SpecialtyID integer,

SpecialtyName varchar(50),

CONSTRAINT PK\_PhysicianSpecialties PRIMARY KEY (SpecialtyID))

go

CREATE TABLE ZipCodes

(ZipCode varchar(10),

City varchar(50),

State varchar(2),

CONSTRAINT PK\_ZipCodes PRIMARY KEY (ZipCode))

go

CREATE TABLE PhysicianPractices

(PracticeID integer,

PracticeName varchar(50),

Address\_Line1 varchar(50),

Address\_Line2 varchar(50),

ZipCode varchar(10),

Phone varchar(14),

Fax varchar(14),

WebsiteURL varchar(50),

CONSTRAINT PK\_PhysicianPractices PRIMARY KEY (PracticeID),

CONSTRAINT FK\_PhysicianPractices\_ZipCodes FOREIGN KEY (ZipCode) REFERENCES Zipcodes)

go

CREATE TABLE Physicians

(PhysicianID integer,

FirstName varchar(40),

LastName varchar(50),

PracticeID integer,

SpecialtyID integer,

Email varchar(50),

CONSTRAINT PK\_Physicians PRIMARY KEY (PhysicianID),

CONSTRAINT FK\_Physicians\_Practices FOREIGN KEY (PracticeID) REFERENCES PhysicianPractices,

CONSTRAINT FK\_Physicians\_PhysicianSpecialities FOREIGN KEY (SpecialtyID) REFERENCES PhysicianSpecialties)

go

CREATE TABLE Patients

(PatientID integer,

FirstName varchar(50),

MiddleInitial varchar(1),

LastName varchar(50),

Address\_Line1 varchar(50),

Address\_Line2 varchar(50),

ZipCode varchar(10),

Phone\_Home varchar(14),

Phone\_Alternate varchar(14),

Email varchar(50),

CONSTRAINT PK\_Patients PRIMARY KEY (PatientID))

go

CREATE TABLE Referrals

(ReferralID integer,

StartDate smalldatetime,

EndDate smalldatetime,

PatientID integer,

PhysicianID integer,

CONSTRAINT PK\_Referrals PRIMARY KEY (ReferralID),

CONSTRAINT FK\_Referrals\_Patients FOREIGN KEY (PatientID) REFERENCES Patients,

CONSTRAINT FK\_Referrals\_Physicians FOREIGN KEY (PhysicianID) REFERENCES Physicians)

go

CREATE TABLE Services

(ServiceID integer,

ServiceName varchar(50),

CONSTRAINT PK\_ServiceID PRIMARY KEY (ServiceID))

go

CREATE TABLE Frequencies

(FrequencyID integer,

Frequency varchar(30),

CONSTRAINT PK\_Frequencies PRIMARY KEY (FrequencyID))

go

CREATE TABLE ReferralServices

(ReferralID integer,

ServiceID integer,

FrequencyID integer,

CONSTRAINT PK\_ReferralServices PRIMARY KEY (ReferralID, ServiceID),

CONSTRAINT FK\_ReferralServices\_Referrals FOREIGN KEY (ReferralID) REFERENCES Referrals,

CONSTRAINT FK\_ReferralServices\_Services FOREIGN KEY (ServiceID) REFERENCES Services,

CONSTRAINT FK\_ReferralServices\_Frequencies FOREIGN KEY (FrequencyID) REFERENCES Frequencies)

go

CREATE TABLE PaymentTypes

(PaymentTypeID integer,

PaymentType varchar(25),

CONSTRAINT PK\_PaymentTypes PRIMARY KEY (PaymentTypeID))

go

CREATE TABLE InsuranceCompanies

(InsuranceID integer,

InsuranceCompany varchar(50),

Address\_Line1 varchar(50),

Address\_Line2 varchar(50),

ZipCode varchar(10),

Phone varchar(15),

Fax varchar(15),

Email varchar(50),

CONSTRAINT PK\_InsuranceCompanies PRIMARY KEY (InsuranceID),

CONSTRAINT FK\_InsuranceCompanies\_ZipCodes FOREIGN KEY (ZipCode) REFERENCES ZipCodes)

go

CREATE TABLE Contracts

(ContractID integer,

ReferralID integer,

StartDate smalldatetime,

EndDate smalldatetime,

PaymentTypeID integer,

InsuranceID integer,

NegotiatedRate float,

CONSTRAINT PK\_Contracts PRIMARY KEY (ContractID),

CONSTRAINT PK\_Contracts\_Referrals FOREIGN KEY (ReferralID) REFERENCES Referrals,

CONSTRAINT FK\_Contracts\_PaymentTypes FOREIGN KEY (PaymentTypeID) REFERENCES PaymentTypes,

CONSTRAINT FK\_Contracts\_InsuranceCompanies FOREIGN KEY (InsuranceID) REFERENCES InsuranceCompanies)

go

CREATE TABLE EmployeeTypes

(EmployeeTypeID integer identity,

EmployeeType varchar(25),

CONSTRAINT PK\_EmployeeTypes PRIMARY KEY (EmployeeTypeID))

go

CREATE TABLE EmployeeTitles

(EmployeeTitleID integer,

EmployeeTitle varchar(30),

CONSTRAINT PK\_EmployeeTitles PRIMARY KEY (EmployeeTitleID))

go

CREATE TABLE EmployeeSkillLevels

(SkillLevelID integer,

SkillLevel varchar(15),

CONSTRAINT PK\_EmployeeSkillLevels PRIMARY KEY (SkillLevelID))

go

CREATE TABLE BillingRates

(EmployeeTypeID integer,

SkillLevelID integer,

BillingRate float,

CONSTRAINT PK\_PrimaryKey PRIMARY KEY (EmployeeTypeID, SkillLevelID),

CONSTRAINT FK\_BillingRates\_EmployeeTypes FOREIGN KEY (EmployeeTypeID) REFERENCES EmployeeTypes,

CONSTRAINT FK\_BillingRates\_EmployeeSkillLevels FOREIGN KEY (SkillLevelID) REFERENCES EmployeeSkillLevels)

go

CREATE TABLE EmployeeRanks

(RankID integer,

EmployeeTypeID integer,

TitleID integer,

SkillLevelID integer,

HourlyRate float,

Salary float,

CONSTRAINT PK\_EmployeeRanks PRIMARY KEY (RankID),

CONSTRAINT FK\_EmployeeRanks\_EmployeeTypes FOREIGN KEY (EmployeeTypeID) REFERENCES EmployeeTypes,

CONSTRAINT FK\_EmployeeRanks\_EmployeeTitles FOREIGN KEY (TitleID) REFERENCES EmployeeTitles,

CONSTRAINT FK\_EmployeeRanks\_EmployeeSkillLevels FOREIGN KEY (SkillLevelID) REFERENCES EmployeeSkillLevels)

go

CREATE TABLE Employees

(EmployeeID integer,

FirstName varchar(30),

MiddleInitial varchar(1),

LastName varchar(50),

Address\_Line1 varchar(50),

Address\_Line2 varchar(50),

ZipCode varchar(10),

Phone varchar(14),

Cell\_Phone varchar(14),

Email varchar(50),

RankID integer,

HourlyWage float,

Salary float,

CONSTRAINT PK\_Employees PRIMARY KEY (EmployeeID),

CONSTRAINT FK\_Employees\_EmployeeRanks FOREIGN KEY (RankID) REFERENCES EmployeeRanks,

CONSTRAINT FK\_Employee\_ZipCodes FOREIGN KEY (ZipCode) REFERENCES ZipCodes)

go

CREATE TABLE Shifts

(ShiftID integer,

ShiftName varchar(20),

StartTime time,

EndTime time,

CONSTRAINT PK\_Shifts PRIMARY KEY (ShiftID))

go

CREATE TABLE DaysOfWeek

(DayOfWeekID integer,

DayOfWeek varchar(15),

CONSTRAINT PK\_DaysOfWeek PRIMARY KEY (DayOfWeekID))

go

CREATE TABLE Availability

(EmployeeID integer,

WeekOf smalldatetime,

DayOfWeekID integer,

ShiftID integer,

CONSTRAINT PK\_Availability PRIMARY KEY (EmployeeID, WeekOf, DayOfWeekID, ShiftID),

CONSTRAINT FK\_Availability\_Employees FOREIGN KEY (EmployeeID) REFERENCES Employees,

CONSTRAINT FK\_Availability\_DaysOfWeek FOREIGN KEY (DayOfWeekID) REFERENCES DaysOfWeek,

CONSTRAINT FK\_Availability\_Shifts FOREIGN KEY (ShiftID) REFERENCES Shifts)

go

CREATE TABLE MedicalSuppliers

(SupplierID integer,

SupplierName varchar(50),

Address\_Line1 varchar(50),

Address\_Line2 varchar(50),

ZipCode varchar(10),

Phone varchar(14),

Fax varchar(14),

Email varchar(50),

CONSTRAINT PK\_MedicalSuppliers PRIMARY KEY (SupplierID),

CONSTRAINT FK\_MedicalSuppliers\_ZipCodes FOREIGN KEY (ZipCode) REFERENCES ZipCodes)

go

CREATE TABLE Supplies

(SupplyID integer,

SupplyDescription varchar(40),

ChargePerUnit float,

CONSTRAINT PK\_Supplies PRIMARY KEY (SupplyID))

go

CREATE TABLE SupplyInventory

(SupplyID integer,

SupplierID integer,

DateReceived smalldatetime,

UnitCost float,

Quantity float,

CONSTRAINT PK\_SupplyInventory PRIMARY KEY (SupplyID, SupplierID, DateReceived),

CONSTRAINT FK\_SupplyInventory\_Supplies FOREIGN KEY (SupplyID) REFERENCES Supplies,

CONSTRAINT FK\_SupplyInventory\_Suppliers FOREIGN KEY (SupplierID) REFERENCES MedicalSuppliers)

go

CREATE TABLE Visits

(VisitID integer,

DateRendered smalldatetime,

StartTime time,

EndTime time,

EmployeeID integer,

PatientID integer,

CONSTRAINT PK\_Visits PRIMARY KEY (VisitID),

CONSTRAINT FK\_Visits\_Employees FOREIGN KEY (EmployeeID) REFERENCES Employees,

CONSTRAINT FK\_Visits\_Patients FOREIGN KEY (PatientID) REFERENCES Patients)

go

CREATE TABLE VisitDetails

(VisitID integer,

VisitDetailID integer,

SupplyID integer,

SupplyQuantity integer,

ServiceID integer,

Charge float,

CONSTRAINT PK\_VisitDetails PRIMARY KEY (VisitID, VisitDetailID),

CONSTRAINT FK\_VisitDetaiils\_Supplies FOREIGN KEY (SupplyID) REFERENCES Supplies,

CONSTRAINT FK\_VisitDetails\_Services FOREIGN KEY (ServiceID) REFERENCES Services)

go

**Step 2: Data Loading**

Load Data properly from Excel File.

**Step 3: Queries**

Write and execute the queries below to answer each question.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Output Format** | **Your Answer:** |
| 1. Display a list of all patients who have a last name beginning with the letter “P”. | Patient Last Name, followed by a comma and a space, followed by the patient’s first name. (e.g. Smith, John)  Sort order: Patient Last Name - ascending | select Lastname + ', ' + FirstName as PatientName  from Patients  where lastname like 'P%'  order by lastname |
| 2. Display a list of all patients who have an alternate/cell phone number | Patient First Name, followed by a space, followed by the patient’s last name. (e.g. Melesa Poole), alternate/cell phone number  Sort order: Patient First Name – ascending  Patient Last Name – ascending | select FirstName + ' ' + LastName as 'Patient Name' from Patients  where Phone\_Alternate is not null  order by FirstName asc, LastName asc |
| 3. Display a list of all patients who do not have an email address. | Patient First Name, followed by a space, followed by the patient’s last name. (e.g. Melesa Poole)  Sort order: Patient First Name – ascending  Patient Last Name – ascending | Select FirstName + ' ' + LastName as 'patient name'  From patients  where Email is NULL  order by firstname asc, LastName asc |
| 4. Display a list of all patients who live in zipcode 24551. | Patient Last Name, Address1, Address2, City, State, Zip  Sort order: Patient Last Name – descending | select P.LastName, P.Address\_Line1, P.Address\_Line2, Z.City, Z.State, P.ZipCode from Patients P  inner join ZipCodes Z on Z.ZipCode = P.ZipCode  where P.ZipCode = '24551'  order by P.LastName desc |
| 5. Display a list of all physicians whose specialty is Internal Medicine or Orthopedics | Physician First Name, space, last name (call this column Physician), Specialty  Sort order: Physician First Name – ascending  Physician Last Name – descending | select FirstName + LastName as 'Physician', SpecialtyName from Physicians p  inner join PhysicianSpecialties ps on p.specialtyid = ps.SpecialtyID  inner join PhysicianPractices pp on pp.PracticeID = p.PracticeID  where SpecialtyName = 'Internal Medicine' or SpecialtyName = 'Orthopedics'  order by p.FirstName asc, p.LastName desc |
| 6. Display a list of all physicians, their specialties, and their practices | Physician Last Name, Specialty, Practice  Sort order: Physician Specialty – ascending  Physician Last Name – ascending  Practice – ascending | select FirstName, LastName, pp.PracticeName, ps.SpecialtyName from Physicians p  inner join PhysicianSpecialties ps on p.specialtyid = ps.SpecialtyID  inner join PhysicianPractices pp on pp.PracticeID = p.PracticeID  order by ps.SpecialtyName asc, LastName asc, PracticeName asc |
| 7. Display a list of all physicians whose practices are in Lynchburg | Physician Last Name, Practice Name, Address, City, State, Zipcode, Phone  Sort order: Zipcode – ascending  Practice Name – descending  Physician Last Name – ascending | select Physicians.LastName, PhysicianPractices.PracticeName,  PhysicianPractices.Address\_Line1, ZipCodes.City, ZipCodes.State, ZipCodes.ZipCode, PhysicianPractices.Phone from Physicians  inner join PhysicianPractices on Physicians.PracticeID = PhysicianPractices.PracticeID  inner join ZipCodes on PhysicianPractices.ZipCode = ZipCodes.ZipCode  where zipcodes.City = 'Lynchburg'  order by zipcode asc, PracticeName Desc, LastName ASC |
| 8. Display the number of physicians in each specialty | Specialty, number of physicians in each specialty  Sort order: Specialty | select SpecialtyName, count(p.SpecialtyID) as 'Number of Physicians' from Physicians p  inner join PhysicianSpecialties ps on p.SpecialtyID = ps.SpecialtyID  group by SpecialtyName  order by SpecialtyName |
| 9. Display the number of physicians in each practice, broken out by specialty | Practice, Specialty, number of physicians in each  Sort order: Practice – ascending  Specialty -- ascending | select PhysicianPractices.PracticeName, PhysicianSpecialties.SpecialtyName,  Count(Physicians.PhysicianID) as 'Number of Physicians'  from Physicians  inner join PhysicianPractices  on Physicians.PracticeID = PhysicianPractices.PracticeID  inner join PhysicianSpecialties  on Physicians.SpecialtyID = PhysicianSpecialties.SpecialtyID  group by PhysicianPractices.PracticeName, PhysicianSpecialties.SpecialtyName  order by PracticeName asc, specialtyName desc |
| 10. Display the list of specialties that have no physicians assigned to them. | Specialty  Sort order: Specialty – ascending | select ps.SpecialtyID, ps.SpecialtyName from PhysicianSpecialties ps  left join Physicians p on ps.SpecialtyID = p.SpecialtyID  Where p.SpecialtyID IS NULL  ORDER BY SpecialtyName ASC |
| 11. Display a list of all referrals whose start date was in 2013. | Patient first name, followed by a space, followed by patient last name (Call this whole field “Patient Name”), Referring Physician Last Name (call this field “Physician”), StartDate, EndDate  Sort Order: StartDate – ascending  Patient First Name – ascending  Physician Last Name - ascending | select p.firstname + ' ' + p.lastname as PatientName, py.firstname + ' ' + py.lastname as Physician from Patients p  inner join Referrals r on r.PatientID = p.PatientID  inner join Physicians py on py.PhysicianID = r.PhysicianID  WHERE YEAR(r.StartDate) = '2013'  ORDER BY StartDate ASC, p.FirstName ASC, py.LastName ASC |
| 12. Display a list of all the referrals whose start date is between October 1, 2014 and November 5, 2014 | Patient first name, followed by a space, followed by patient last name (Call this whole field “Patient Name”), Referring Physician Last Name (call this field “Physician”), StartDate, EndDate  Sort Order: StartDate – ascending  Patient First Name – ascending  Physician Last Name - ascending | select p.firstname + ' ' + p.lastname, ph.lastname as 'Physician', r.startdate, r.enddate from Referrals r  inner join Physicians ph on ph.PhysicianID = r.PhysicianID  inner join Patients p on p.PatientID = r.PatientID  where r.StartDate between '2014-10-01' and '2014-11-05'  order by r.StartDate asc, p.firstname asc, ph.lastname asc |
| 13. Display the number of referrals given by each physician | Physician Last name, Physician First Name, number of referrals  Sort Order: Physician Last Name – ascending  Physician First Name – ascending | select Physicians.LastName, Physicians.FirstName, COUNT(referrals.ReferralID) as 'Number of Referrals'  from Physicians  inner join referrals  on physicians.physicianID = referrals.PhysicianID  group by Physicians.FirstName, Physicians.LastName  order by Physicians.LastName ASC, Physicians.FirstName Asc |
| 14. List the number of referrals in 2014 for each service requested. | Service name, number of referrals  Sort order: Service name | select s.ServiceName, count(r.ReferralID) as 'Number of referrals' from referralservices rs  inner join services s on s.serviceid = rs.ServiceID  inner join Referrals r on r.ReferralID = rs.ReferralID  where YEAR(r.StartDate) = 2014 and YEAR(r.EndDate) = 2014  group by s.ServiceName  order by s.ServiceName |
| 15. Display a list of all patients requiring exercise therapy in 2013 | Patient Last Name, Patient First Name  Sort order: Patient last name – ascending  Patient first name – ascending | select distinct P.Lastname, P.FirstName from Patients P  inner join Referrals R on R.PatientID = P.PatientID  inner join ReferralServices RS on RS.ReferralID = R.ReferralID  inner join Services S on S.ServiceID = RS.ServiceID  where ServiceName = 'Exercise Therapy' and year(R.StartDate) = 2013  order by P.LastName asc, P.FirstName asc |
| 16. Display a list of any referrals that require “Insulin injections” and “2x Daily” is NOT listed as their frequency. | Patient Last Name, Physician Last Name, referral start date  Sort order: Physician Last Name – ascending  Patient Last Name – ascending  Referral Start Date – ascending | select p.lastname, ph.lastname, r.startdate from Referrals r  inner join Patients p on p.PatientID = r.PatientID  inner join Physicians ph on ph.PhysicianID = r.PhysicianID  inner join ReferralServices rs on rs.ReferralID = r.ReferralID  inner join Services s on s.ServiceID = rs.ServiceID  inner join Frequencies f on f.FrequencyId = rs.FrequencyID  where f.Frequency <> '2X Daily' and s.ServiceName = 'Insulin Injections'  order by ph.LastName asc, p.LastName asc, r.StartDate asc |
| 17. Display the contracts and payment methods associated with each referral | Patient Last Name, Physician Last Name, Referral Start Date, Contract Start Date, Payment Method  Sort Order: Payment Method - ascending  Physician Last Name – ascending  Patient Last Name – ascending  Referral Start Date – ascending  Contract Start Date – ascending | select p.LastName, py.LastName, r.startdate, c.StartDate, pt.paymenttype from Patients p  inner join Referrals r on r.PatientID = p.PatientID  inner join Contracts c on c.ReferralID = r.ReferralID  inner join PaymentTypes pt on pt.PaymentTypeID = c.PaymentTypeID  inner join Physicians py on py.PhysicianID = r.PhysicianID  ORDER BY pt.PaymentType asc, py.LastName ASC, p.LastName ASC, r.StartDate ASC, c.StartDate ASC |
| 18. Display the number of contracts whose payment method is Insurance | Number of contracts (This is a single value) | select count(c.PaymentTypeID) from Contracts c  inner join PaymentTypes pt on pt.PaymentTypeid = c.PaymentTypeID  where pt.PaymentType = 'Insurance' |
| 19. Display the number of contracts whose payment method is Insurance, broken out by Insurance Company | Insurance Company Name, number of contracts  Sort order: Insurance company name | select InsuranceCompany, count(c.contractid) from Contracts c  inner join InsuranceCompanies ic on c.InsuranceID = ic.InsuranceID  inner join PaymentTypes pt on pt.PaymentTypeID = c.PaymentTypeID  group by InsuranceCompany  order by InsuranceCompany |
| 20. List the Employees who are Nurses | Employee First Name, followed by a space, followed by Employee Middle Initial, followed by a space, followed by Employee Last Name (call this whole field “Nurses”) | select e.firstname + ' ' + ISNULL(e.MiddleInitial, '') + ' ' + e.lastname as 'Nurses' from EmployeeRanks er  inner join Employees e on e.RankID = er.RankID  inner join EmployeeTypes et on et.EmployeeTypeID = er.EmployeeTypeID  where EmployeeType = 'Nurse' |
| 21. Display the average hourly wage for all employees who are aides. | Average hourly wage (single value) | select AVG(er.HourlyRate) as AverageWage from EmployeeRanks er  inner join EmployeeTypes et on et.EmployeeTypeID = er.EmployeeTypeID  WHERE et.EmployeeType = 'Aide' |
| 22. Display the average hourly wage for all hourly employees broken out by level. | Skill level, average wage  Sort order: Skill Level | select ESL.SkillLevel, avg(E.HourlyWage) as 'Average Wage' from Employees E  inner join EmployeeRanks ER on ER.RankID = E.RankID  inner join EmployeeSkillLevels ESL on ESL.SkillLevelID = ER.SkillLevelID  group by ESL.SkillLevel  order by esl.SkillLevel |
| 23. Display the total salary for all salaried employees. | Total salaries (single value) | select SUM(e.Salary) as 'Total Salaries' from Employees e  Where e.Salary IS NOT NULL |
| 24. Display the number of employees assigned to each rank. | RankID, Employee Type, Skill Level, Employee Title, number of employees  Sort Order: RankID – ascending  Employee type – ascending  Skill Level – ascending  Employee Title – ascending | select er.RankID, EmployeeType, SkillLevel, EmployeeTitle, count(er.RankID) as 'Number of Employees' from EmployeeRanks er  inner join EmployeeTitles et on et.EmployeeTitleID = er.TitleID  inner join Employees e on e.RankID = er.RankID  inner join EmployeeSkillLevels esl on esl.SkillLevelID = er.SkillLevelID  inner join EmployeeTypes ety on ety.EmployeeTypeID = er.EmployeeTypeID  group by er.RankID, ety.EmployeeType, esl.SkillLevel, et.EmployeeTitle  order by RankID asc, employeetype asc, skilllevel asc, EmployeeTitle asc |
| 25. Display a list of Employees who are nurses and were available to work on Sunday evenings during the week of 11/2/2014 | Employee Last Name, Employee First Name  Sort order: Last Name – ascending  First Name – ascending | select e.LastName, e.FirstName from EmployeeRanks er  inner join EmployeeTitles etit on etit.EmployeeTitleID = er.EmployeeTypeID  inner join EmployeeTypes et on et.EmployeeTypeID = er.EmployeeTypeID  inner join Employees e on e.RankID = er.RankID  where et.EmployeeType = 'Nurse'  and EmployeeID in(  select EmployeeID from Availability a  inner join DaysOfWeek d on d.DayOfWeekID = a.DayOfWeekID  inner join Shifts s on s.ShiftID = a.ShiftID  where d.DayOfWeek = 'Sunday' and a.WeekOf = '11/2/2014' and s.ShiftName = 'Evening')  order by e.LastName asc, e.FirstName asc |
| 26. Display a list of Employees who were available to work during morning shifts during the week of 11/2/2014 and had a skill level of level 3. | Employee Last Name, Employee First Name,  Employee Type, Employee Title  Sort order: Employee Type – ascending  Employee Title – ascending  Employee Last Name – ascending  Employee First Name – ascending | select e.LastName, e.FirstName, et.EmployeeType, etit.EmployeeTitle from EmployeeRanks er  inner join EmployeeTitles etit on etit.EmployeeTitleID = er.EmployeeTypeID  inner join EmployeeTypes et on et.EmployeeTypeID = er.EmployeeTypeID  inner join Employees e on e.RankID = er.RankID  where er.SkillLevelID = 3 and EmployeeID in(  select EmployeeID from Availability a  inner join Shifts s on s.ShiftID = a.ShiftID  where a.WeekOf = '11/2/2014' and s.ShiftName = 'Morning')  order by et.EmployeeType asc, etit.EmployeeTitle asc, e.LastName asc, e.FirstName asc |
| 27. Display the total quantity of catheters added to inventory during 2013. | Total catheters (single value) | select sum(Quantity) as 'Total catheters' from SupplyInventory  where SupplyID = (select supplyID from Supplies where SupplyDescription = 'catheters')  and year(DateReceived) = 2013 |
| 28. Display the total cost of “sterile gloves – small” provided by Poole’s Medical supplies during 2013. | Total cost (single value) | select sum(unitcost \* quantity) from SupplyInventory  where year(DateReceived) = 2013 and SupplyID in(  select SupplyID from Supplies where SupplyDescription = 'sterile gloves - small')  and supplierid in(select SupplierID from MedicalSuppliers where SupplierName = 'Pooles Medical Supplies') |
| 29. Display the average cost of supplies for each supply item broken out by supplier. | Supply, Supplier, Average cost per supply item  Sort order: Supply – ascending  Supplier – ascending | Select s.supplyDescription, m.SupplierName, AVG(si.UnitCost) as 'Average Cost' from Supplies s  inner join SupplyInventory si on si.SupplyID = s.SupplyID  inner join MedicalSuppliers m on m.SupplierID = si.SupplierID  group by s.SupplyDescription, m.SupplierName  Order by s.SupplyDescription ASC, m.SupplierName ASC |
| 30. Display the total cost of all items purchased from suppliers broken out by supplier. | Supplier, Total cost of all items provided by supplier  Sort order: Supplier – ascending | select ms.suppliername, SUM(unitcost \* quantity) from SupplyInventory si  inner join MedicalSuppliers ms on si.SupplierID = ms.SupplierID  where si.SupplierID in(  select SupplierID from SupplyInventory)  group by ms.SupplierName  order by ms.SupplierName asc |
| 31. Display a list of all the visits that occurred from March 20, 2014 to March 25, 2014 (including March 20 and March 25) | DateRendered, Patient Last Name, Employee Last Name, Start Time, End time  Sort order: DateRendered – ascending  Patient Last Name – ascending  Employee Last Name – ascending  Start Time – ascending | select v.DateRendered, p.LastName, e.LastName, v.StartTime, v.EndTime from visits v  inner join Patients p on p.PatientID = v.PatientID  inner join Employees e on e.EmployeeID = v.EmployeeID  WHERE v.DateRendered BETWEEN '2014-03-20' AND '2014-03-25'  order by v.DateRendered asc, p.LastName asc, e.LastName asc, v.StartTime asc |
| 32. List the total charges for the visit that occurred on 2/12/2014 for Helen Ramirez that was provided by Laura White. | Total charges (single value) | select sum(charge) from Visits v  inner join VisitDetails vd on vd.VisitID = v.VisitID  inner join Employees e on e.EmployeeID = v.EmployeeID  inner join Patients p on p.PatientID = v.PatientID  where v.DateRendered = '2014-02-12' and e.FirstName ='Laura' and e.LastName = 'White' and p.FirstName = 'Helen' and p.LastName = 'Ramirez' |
| 33. List the number of patients who received insulin injections during 2014 (Note this is the number of unique patients who ever received insulin injections – not the number of visits in which insulin injections were provided). | Total number of patients (single value) | select distinct count(PatientID) from Patients  where PatientID in(  select PatientID from Visits  where VisitID in(  select VisitID from VisitDetails  where serviceID in(  select serviceID from Services  where ServiceName = 'Insulin Injections'))) |
| 34. List the total number of 4” self-adhesive bandages that were used in 2014 | Total number of 4” self-adhesive bandages (single value) | select count(supplyquantity) from VisitDetails vd  inner join Supplies s on s.supplyId = vd.SupplyID  where s.SupplyDescription = '4" self-adhesive bandages' and visitid in(  select visitID from Visits  where year(daterendered) = 2014) |
| 35. List the average charge per visit per month in 2014 broken out by months | Month, average cost per visit  Sort order: month number - ascending | select month(V.DateRendered) as 'Month', avg(VD.Charge) as 'Average cost per visit' from VisitDetails VD  inner join Visits V on V.VisitID = VD.VisitID  where year(V.DateRendered) = 2014  group by month(V.DateRendered)  order by Month |
| 36. Provide a unique list of patients who received visits for feeding from November 1, 2014 until the current date. | Patient Last Name, Patient First Name  Sort order: Patient Last Name – ascending  Patient First Name – ascending | select distinct LastName, FirstName from Patients  inner join visits  on Visits.PatientID = Patients.PatientID  inner join VisitDetails  on VisitDetails.VisitID = Visits.VisitID  inner join services  on services.ServiceID = VisitDetails.ServiceID  where Services.ServiceName = 'feeding' and  Visits.DateRendered between '11/1/2014' and getdate()  group by LastName, FirstName  order by LastName ASC, FirstName ASC |