|  |
| --- |
| CS 585 Assignment- 2 solutions |
| Shashank Gowdagiri |
| The following are the solutions to the second assignment of the CS 585 class due on October 8, 2013. |

10/8/2013

1. Solutions:

Patrol has an ID which is the index for Designation, License plate, and Date and Time. This needs to be referenced by the Conducts key, which is a relation from the police officers’ primary key, Badge Number.

Precinct (Station No, Address, Badge No)

[N] [S] [N]

* From Precinct.*Badge No* to Senior Officer.Badge No

Jurisdiction (Designation, Station No)

[S] [N]

* From Jurisdtcition.*Designation* to Zone.Designation
* From Jurisdiction.*Station No* to Precinct.Station No

Works\_in (Badge no, Station No)

[N] [N]

* From Works\_in.*Badge No* to Police Officer.Badge No
* From Works\_in.*Station No* to Precinct.Station No

Police Officer (Badge No, Name, Rank)

[N] [S] [S]

Senior Officer (Badge No)

[N]

* From Senior Officer.*Badge No* to Police Officer.Badge No

Detective (Badge No)

[N]

* From Detective.*Badge No* to Police Officer.Badge No

(The next relation links the detectives who are a part of the Investigation Team)

Member of (Badge No, Case No)

[N] [S]

* From Member Of.*Badge No* to Detective.Badge No
* From Member Of. *Case No* to Investigation Team.Case No

Senior Detective (Badge No, Case No)

[N] [S]

* From Senior Detective*.Badge No* to Senior Officer.Badge No
* From Senior Detective*.Badge No* to Detective.Badge No
* From Senior Detective*.Case No* to Investigation.Case No

Patrol\_Zone\_Of (Designation, PID)

[S] [N]

* From Patrol\_Zone\_Of.*PID* to Patrol.PID

Patrol (PID, Date and Time, License Plate, Designation)

[N] [T] [N] [S]

Conducts (Badge No, PID)

[N] [N]

* From Conducts.*Badge No* to Police Officer.Badge No
* From Conducts.*PID* to Patrol.PID

Vehicle (License Plate, Shop No)

[N] [N]

Zone (Designation, Neighborhood, Station No)

[S] [S] [N]

Investigation Team (Case No, LBadge No)

[S] [N]

* From Investigation Team.*LBadge No* to Senior Detective.Badge

The precinct commander and the senior officer classes were designed to optimize and reduce the table size. The tradeoff is that the senior officer has a lot of dependencies, but it would be relatively easier to find a police officer with the ranking of a senior officer. In case of the rank of the commander of the Precinct, he just followed the badge ID of the senior officer, whose class derives its Badge Id from the Police officer class

1. A)

Specify the following queries in relational algebra:

1. List the full details for all service types that take more than two days to complete.

Service\_more\_than\_2 ← σ(Service Type.HoursRequired>48) (Service Type)

Result ← π (ServiceID, Name, PartsCost,) (Service\_more\_than\_2)

1. Give the first and last names and the hourly rate of all electricians who worked on a Toyota Camry between 2013-08-01 and 2013-08-31.

Date\_Mech ← π (Mechanic)σ(Work\_Order.Date>=2013-08-01 ∧ Work\_Order.Date<=2013-08-31 ) (Work\_Order)

Camry← σ (Car.Make=”Toyota” ∧ Car.Model=”Camry”) (Car)

Mech\_dude ← Person\*Mechanic

Result ← π (Mechanic.First Name, Mechanic.Last Name, Mechanic.Specialty) (Camry ⋈(Work\_Order.Car=Car.VIN)Date\_Mech ⋈ Date\_mech.Mechanic=mech\_dude.SSN ^ mech\_dude.speciality=” Electrician” Mech\_dude)

1. Find all mechanic-owned cars that were fixed by their own owner in the shop. Give the make and model of each car, along with the address and phone number of the owner.

Work\_Car ← (WORK\_ORDER ⋈ WORK\_ORDER .Car=Car.Vin Car)

Work\_Order\_stuff ← (Work\_Car ⋈ (Work\_car.OrderNo= WORK\_ORDER\_ITEM.ORDERNO) ^ (WORK\_ORDER\_ITEM.Mechanic=Work\_car.Owner) Work\_Order\_Item)

Customer\_Person ← (Person ⋈ Customer)

Customer\_stuff← Customer\_Person ⋈ Customer\_Person.SSN=Work\_Order\_stuff.Owner Work\_Order\_stuff

Res← π(make,model,address,phonenumber) Customer\_stuff

iv. List the order numbers and dates of all work orders involving a Nissan Versa in which all of the requested services cost less than $100 in parts.

Work\_Order\_stuff← Work\_Order ⋈ Work\_Order.OrderNo = Work\_Order\_Item.OrderNo Work\_Order\_Item

Car\_stuff ← Car ⋈ Car.Vin=Work\_order\_stuff.Car Work\_Order\_stuff

Car\_service ← Car\_stuff ⋈ Car\_stuff.Service=Service\_Type.ServiceID Service\_Type Service\_Type

Mech\_Car ← Mechanic ⋈ Mechanic.SSN=Car\_service.SSN  Car\_service

Result ← π (OrderNo,Date) (σ((make=”Nissan”^Model=”Versa”) ^ (Partscost+ (Hoursrequired\*hourlyrate)<100)) Mech\_Car

B)

i) T2

K L M

25 q 7

20 q 6

ii)

C-M = {2,6,4}-{4,7,6,3}

2

iii)

A B C

50 r 2

20 q 6

30 s 4

50 s 4

25 q 7

20 q 6

60 t 3

iv)

A B C K L M

50 r 2 60 t 4

30 s 4 50 s 4

30 s 4 60 t 3

1. Relational Schema:

Legend:

N – number

S – string

Ty- type (“classroom”, “conference room”, “activity room”, “study room”)

T – Time

B – Yes/No

Assertion: If there is a room number, capacity and type, in a building, then the room’s entry and exit access is between the opening and closing times of that specified building.

The room number is represented in the request.

The amenities which are provided are represented in strings.

Building (Name, Opening time, Closing Time)

[S] [T] [T]

Room (RoomNo, BName , RoomType, Capacity, RoomEntryAccess,RoomExitAccess)

[N] [S] [Ty] [N] [T] [T]

* From Room.*Bname* to Building.Name

Amenities (RoomNo,Provided)

[N] [S]

* From Amenities.*RoomNo* to Room.RoomNo

Requester (REmail, RName)

(RName is not a primary key because there can be multiple people with the same name, but every requester’s email ID is always unique)

Request (RequestRoomNo, ReqEmail, RStart\_time, Rend\_Time, RCapacity, Accesss)

[N] [S] [T] [T] [N] [B]

* From Request.*ReqEmail* to Requester.REmail
* From Request.*RequestRoomNo* to Room.RoomNo
* From Request.*Rstart\_time* to Room.RoomEntryAccess

(This design for the schema does not take into account the application level checking for the request).