**Comp 353 Section F**

**Main project**

**Fall 2021**

**Report**

**Group Account: mnc353\_2**

**Peng Zhao 40070966**

**Zehui Cao 40044803**

**Gu JiaQi 40082054**

**Zhen Han 40054003**

**Concordia University**

**Prof Khaled Jababo**

**Section F**

**2021.12.1**

**Faculty of Engineering and Computer Science**

**Expectations of Originality**

**This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering**

**and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and**

**take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct.**

**The course outline may stipulate additional requirements for the course.**

**1. Your submissions must be your own original work. Group submissions must be the original work of the**

**students in the group.**

**2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks,**

**and must be attributed to the source by a numerical reference citation1. Note that engineering reports**

**rarely contain direct quotations.**

**3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference**

**citation.**

**4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site**

**by numerical reference citation.**

**5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that**

**source by a numerical reference citation.**

**6. No part of any assignment, lab report or project report submitted for this course can be submitted for**

**any other course.**

**7. In preparing your submissions, the work of other past or present students cannot be consulted, used,**

**copied, paraphrased or relied upon in any manner whatsoever.**

**8. Your submissions must consist entirely of your own or your group’s ideas, observations, calculations,**

**information and conclusions, except for statements attributed to sources by numerical citation.**

**9. Your submissions cannot be edited or revised by any other student.**

**10. For lab reports, the data must be obtained from your own or your lab group’s experimental work.**

**11. For software, the code must be composed by you or by the group submitting the work, except for code**

**that is attributed to its sources by numerical reference.**

**You must write one of the following statements on each piece of work that you submit:**

**For individual work: “I certify that this submission is my original work and meets the Faculty's**

**Expectations of Originality”, with your signature, I.D. #, and the date.**

**For group work: “We certify that this submission is the original work of members of the group and meets**

**the Faculty's Expectations of Originality”, with the signatures and I.D. #s of all the team members and the**

**date.**

**A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.**

**I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I**

**certify that all the work I will submit for this course will comply with these requirements and with additional**

**requirements stated in the course outline.**

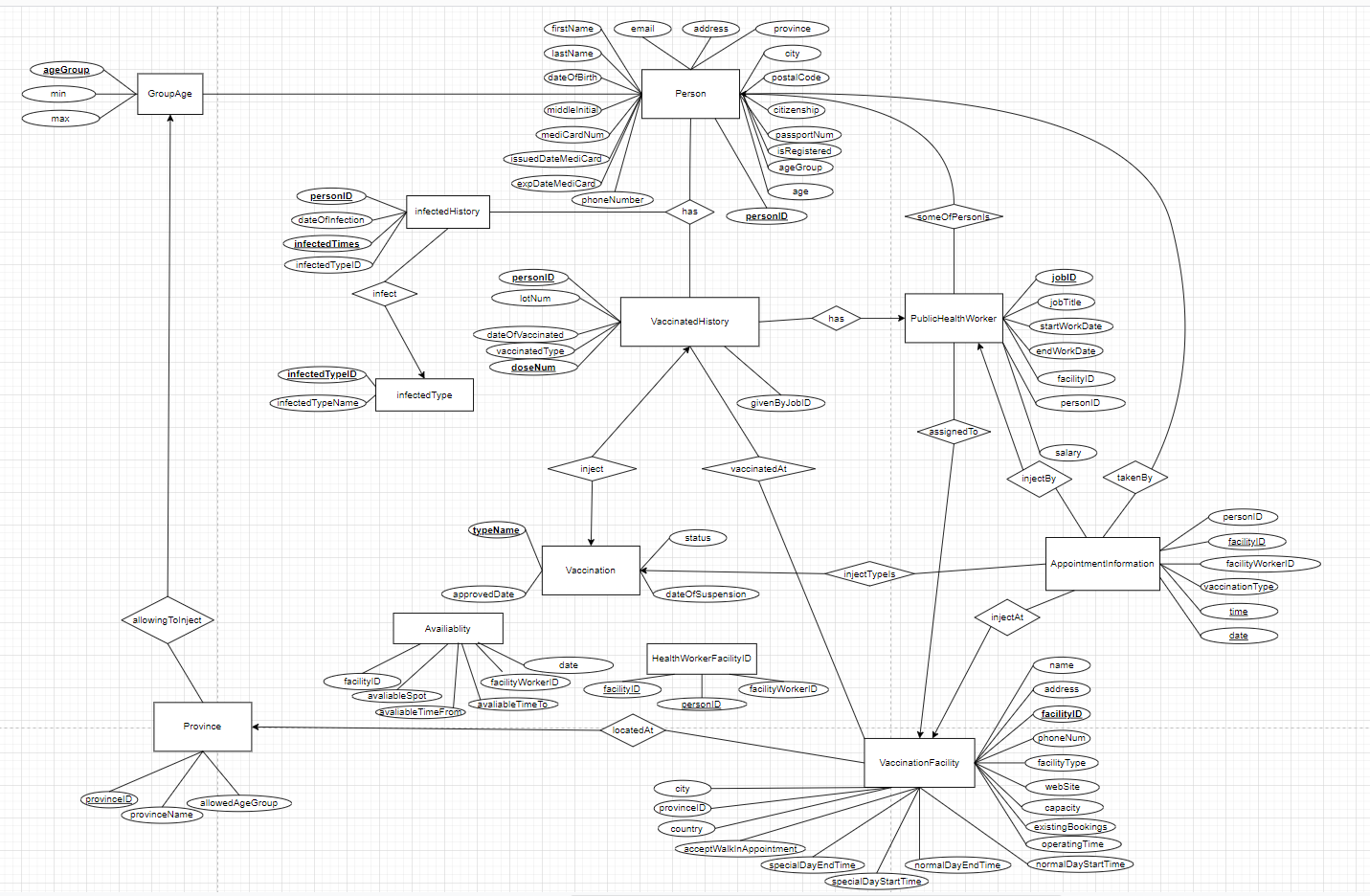
**Course Number: \_Comp 353 Section F\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Instructor: \_\_Khaled Jababo\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name: Peng Zhao, \_Zehui Cao\_\_\_,Gu JiaQi\_\_\_\_\_Zhen Han\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ I.D. # 40070966\_\_, 40044803\_\_\_\_\_\_\_,40082054\_\_\_40054003\_\_\_\_\_\_\_\_\_**

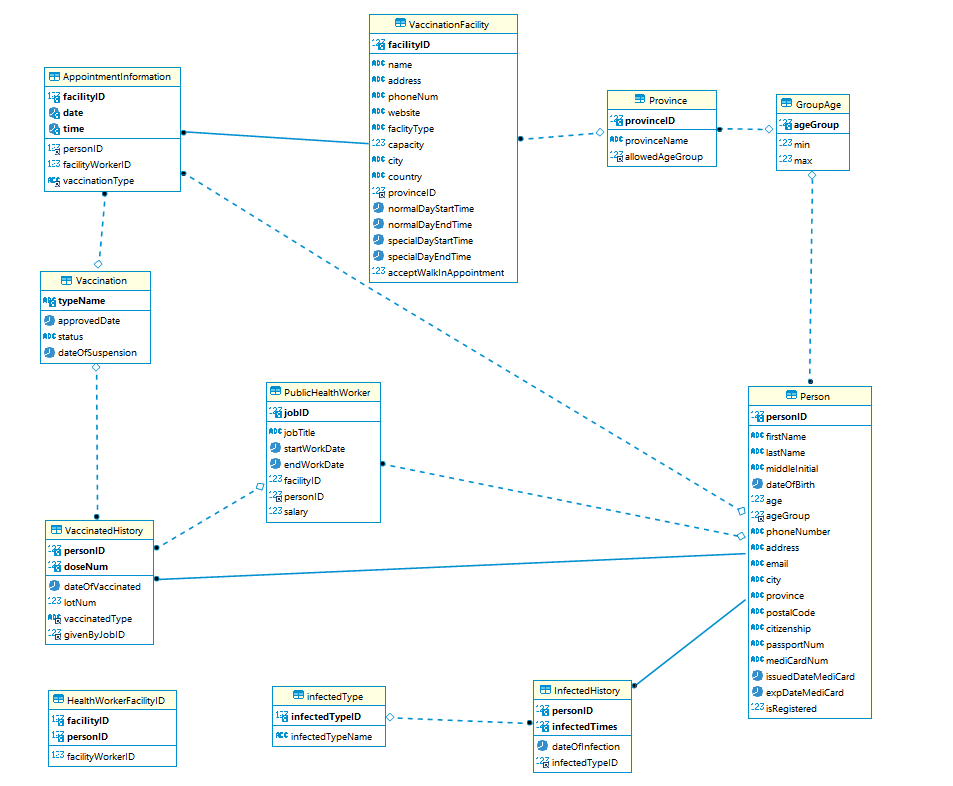
**Signature: Peng\_Zehui Cao ,\_\_\_zhen han\_\_\_\_\_\_\_\_\_, Gu JiaQi\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_2021.12.2\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1.E/R diagram and relation schemas:**

1. **E/R diagram**

****

**（2）Relation schemas:**

****

**3.Relation schemas, FDs:**

**Persons(personID,firstName,lastName,middleInitial,dateOfBirth,age,ageGroup,phoneNumber,address,email,city,province,postalCode,citizenship,passportNum,mediCardNum,issuedDateMediCard,expDateMediCard,isRegistered)**

personID->firstName,lastName,middleInitial,dateOfBirth,age,ageGroup,phoneNumber,address,email,city,province,postalCode,citizenship,passportNum,mediCardNum,issuedDateMediCard,expDateMediCard,isRegistered

passportNum->personID

mediCardNum->personID

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

GroupAge(ageGroup,min,max)

ageGroup->min,max

min,max->ageGroup

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

Province(provinceID,provinceName,allowedAgeGroup)

provinceID->provinceName,allowedAgeGroup

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

InfectedHistory(personID,infectedTimes,dateOfInfection,infectedTypeID)

personID,infectedTimes->dateOfInfection,infectedTypeID

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

InfectedType(infectedTypeID,infectedTypeName)

infectedTypeID->infectedTypeName

infectedTypeName->infectedTypeID

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

VaccinationFacility(facilityID,name,address,phoneNum,website,faclityType,capacity,city,country,provinceID,normalDayStartTime,normalDayEndTime,specialDayStartTime,specialDayEndTime,acceptWalkInAppointment)

facilityID->name,address,phoneNum,website,faclityType,capacity,city,country,provinceID,normalDayStartTime,normalDayEndTime,specialDayStartTime,specialDayEndTime,acceptWalkInAppointment

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

PublicHealthWorker(jobTitle,startWorkDate,endWorkDate,jobID,facilityID,personID,salary)

jobID->jobTitle,startWorkDate,endWorkDate,facilityID,personID,salary

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

HealthWorkerFacilityID(facilityID,personID,facilityWorkerID)

facilityID,personID->facilityWorkerID

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

AppointmentInformation(facilityID,personID,facilityWorkerID,vaccinationType,date,time)

facilityID,date,time->personID,facilityWorkerID,vaccinationType

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

Vaccination(typeName,approvedDate,status,dateOfSuspension)

typeName->approvedDate,status,dateOfSuspension

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

VaccinatedHistory(personID,dateOfVaccinated,doseNum,lotNum,vaccinatedType,givenByJobID)

personID,doseNum->dateOfVaccinated,lotNum,vaccinatedType,givenByJobID

It's in BCNF since for every non trivial FD, the left-hand is a superkey .

**2.Create table Script:**

**CREATE TABLE `AppointmentInformation` (**

**`facilityID` int NOT NULL,**

**`personID` int NOT NULL,**

**`vaccinationType` char(255) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`date` date NOT NULL,**

**`time` time NOT NULL,**

**`timeTo` time DEFAULT NULL,**

**PRIMARY KEY (`facilityID`,`time`,`date`),**

**KEY `taken\_by` (`personID`),**

**KEY `vaccine\_type\_is` (`vaccinationType`),**

**CONSTRAINT `plan\_to\_Inject\_At` FOREIGN KEY (`facilityID`) REFERENCES `VaccinationFacility` (`facilityID`),**

**CONSTRAINT `taken\_by` FOREIGN KEY (`personID`) REFERENCES `Person` (`personID`),**

**CONSTRAINT `vaccine\_type\_is` FOREIGN KEY (`vaccinationType`) REFERENCES `Vaccination` (`typeName`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `Avaliablity` (**

**`facilityID` int DEFAULT NULL,**

**`facilityWorkerID` int DEFAULT NULL,**

**`avaliableSpot` varchar(255) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`avaliableTimeFrom` time DEFAULT NULL,**

**`avaliableTimeTo` time DEFAULT NULL,**

**`date` date DEFAULT NULL**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `GroupAge` (**

**`ageGroup` int NOT NULL,**

**`min` int DEFAULT NULL,**

**`max` int DEFAULT NULL,**

**PRIMARY KEY (`ageGroup`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `HealthWorkerFacilityID` (**

**`facilityID` int NOT NULL,**

**`personID` int NOT NULL,**

**`facilityWorkerID` int DEFAULT NULL,**

**PRIMARY KEY (`facilityID`,`personID`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `InfectedHistory` (**

**`personID` int NOT NULL,**

**`infectedTimes` int NOT NULL DEFAULT '0',**

**`dateOfInfection` date DEFAULT NULL,**

**`infectedTypeID` int DEFAULT NULL,**

**UNIQUE KEY `Infected\_record` (`personID`,`infectedTimes`),**

**KEY `type\_of\_infection` (`infectedTypeID`),**

**CONSTRAINT `from\_The\_Person` FOREIGN KEY (`personID`) REFERENCES `Person` (`personID`),**

**CONSTRAINT `type\_of\_infection` FOREIGN KEY (`infectedTypeID`) REFERENCES `infectedType` (`infectedTypeID`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `Person` (**

**`personID` int NOT NULL,**

**`firstName` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,**

**`lastName` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,**

**`middleInitial` varchar(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,**

**`dateOfBirth` date NOT NULL,**

**`age` int NOT NULL,**

**`ageGroup` int NOT NULL,**

**`phoneNumber` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,**

**`address` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`email` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,**

**`city` varchar(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`province` varchar(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`postalCode` varchar(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`citizenship` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`passportNum` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`mediCardNum` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`issuedDateMediCard` date DEFAULT NULL,**

**`expDateMediCard` date DEFAULT NULL,**

**`isRegistered` int DEFAULT NULL,**

**PRIMARY KEY (`personID`),**

**KEY `belongs\_to\_age\_group` (`ageGroup`),**

**CONSTRAINT `belongs\_to\_age\_group` FOREIGN KEY (`ageGroup`) REFERENCES `GroupAge` (`ageGroup`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `Province` (**

**`provinceID` int NOT NULL,**

**`provinceName` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`allowedAgeGroup` int DEFAULT NULL,**

**PRIMARY KEY (`provinceID`),**

**KEY `current\_age\_group` (`allowedAgeGroup`),**

**CONSTRAINT `current\_age\_group` FOREIGN KEY (`allowedAgeGroup`) REFERENCES `GroupAge` (`ageGroup`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `PublicHealthWorker` (**

**`jobTitle` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`startWorkDate` date DEFAULT NULL,**

**`endWorkDate` date DEFAULT NULL,**

**`jobID` int NOT NULL,**

**`facilityID` int DEFAULT NULL,**

**`personID` int NOT NULL,**

**`salary` int DEFAULT NULL,**

**PRIMARY KEY (`jobID`),**

**KEY `the\_person\_info` (`personID`),**

**CONSTRAINT `the\_person\_info` FOREIGN KEY (`personID`) REFERENCES `Person` (`personID`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `VaccinatedHistory` (**

**`personID` int NOT NULL,**

**`dateOfVaccinated` date DEFAULT NULL,**

**`doseNum` int NOT NULL,**

**`lotNum` int DEFAULT NULL,**

**`vaccinatedType` char(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`givenByJobID` int DEFAULT NULL,**

**PRIMARY KEY (`personID`,`doseNum`),**

**KEY `type\_of\_vaccine` (`vaccinatedType`),**

**KEY `takening\_from` (`givenByJobID`),**

**CONSTRAINT `takening\_by` FOREIGN KEY (`personID`) REFERENCES `Person` (`personID`),**

**CONSTRAINT `takening\_from` FOREIGN KEY (`givenByJobID`) REFERENCES `PublicHealthWorker` (`jobID`),**

**CONSTRAINT `type\_of\_vaccine` FOREIGN KEY (`vaccinatedType`) REFERENCES `Vaccination` (`typeName`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `Vaccination` (**

**`typeName` varchar(255) COLLATE utf8\_unicode\_ci NOT NULL,**

**`approvedDate` date DEFAULT NULL,**

**`status` char(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`dateOfSuspension` date DEFAULT NULL,**

**PRIMARY KEY (`typeName`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `VaccinationFacility` (**

**`facilityID` int NOT NULL,**

**`name` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`address` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`phoneNum` char(30) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`website` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`faclityType` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`capacity` int DEFAULT NULL,**

**`city` varchar(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT 'Montreal',**

**`country` char(255) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci DEFAULT 'Canada',**

**`provinceID` int DEFAULT '1',**

**`normalDayStartTime` timestamp NULL DEFAULT NULL,**

**`normalDayEndTime` timestamp NULL DEFAULT NULL,**

**`specialDayStartTime` timestamp NULL DEFAULT NULL,**

**`specialDayEndTime` timestamp NULL DEFAULT NULL,**

**`acceptWalkInAppointment` tinyint(1) DEFAULT NULL,**

**PRIMARY KEY (`facilityID`),**

**KEY `from\_The\_Province` (`provinceID`),**

**CONSTRAINT `from\_The\_Province` FOREIGN KEY (`provinceID`) REFERENCES `Province` (`provinceID`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**CREATE TABLE `infectedType` (**

**`infectedTypeID` int NOT NULL,**

**`infectedTypeName` char(255) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**PRIMARY KEY (`infectedTypeID`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci**

**3. Question 1~20.**

**//To-DO**

**1.**

**// create new person**

**insert INTO person(personID,firstName,lastName,middleInitial,dateOfBirth,**

**age,ageGroup,phoneNumber,address,email,city,province,postalCode,**

**citizenship,passportNum,mediCardNum,issuedDateMediCard,expDateMediCard,isRegistered)**

**values ('$personID','$firstName','$lastName','$middleInitial','$birth','$age','$ageGroup','$phone','$address','$email'**

**,'$city','$province','$postalCode','$citizenship','$passport','$mediCard','$issuedDate','$expDate','$isRegistered');";**

**// edit person**

**update person set firstName = '$firstName',lastName = '$lastName',middleInitial = '$middleInitial',dateOfBirth = '$birth',**

**age = '$age',ageGroup = '$ageGroup',phoneNumber = '$phone',address = '$address',email = '$email',city = '$city',province = '$province',postalCode = '$postalCode',**

**citizenship = '$citizenship',passportNum = '$passport',mediCardNum = '$mediCard',issuedDateMediCard = '$issuedDate',expDateMediCard = '$expDate',isRegistered = '$isRegistered'**

**where person.personID = '$personID';**

**// display person**

**select \***

**from person**

**where person.personID = '$id'**

**// delete person**

**Delete from person where person.personID = '$personID'**

**2.**

**Create a Public Health Worker:**

**insert INTO PublicHealthWorker(jobTitle,startWorkDate,endWorkDate,jobID,facilityID,personID,salary)**

**values ('$jobTitle','$startWorkDate','$endWorkDate','$jobID','$facilityID','$personID','$salary');**

**Delete a Public Health Worker:**

**Delete from PublicHealthWorker where jobID = '$jobID'**

**Edit a Public Health Worker:**

**update PublicHealthWorker set jobTitle = '$jobTitle',startWorkDate = '$startWorkDate',endWorkDate = '$endWorkDate',jobID = '$jobID',facilityID = '$facilityID',personID = '$personID',salary = '$salary'**

**where PublicHealthWorker.jobID = '$oldJobID';**

**Display a Public Health Worker:**

**select \* from PublicHealthWorker where PublicHealthWorker.jobID = '$id'**

**3.**

**Create a Public Health Facility:**

**insert INTO VaccinationFacility(facilityID,name,address,phoneNum,website,faclityType,capacity,city,country,provinceID,normalDayStartTime,normalDayEndTime,specialDayStartTime,specialDayEndTime,acceptWalkInAppointment)**

**values ('$facilityID','$name','$address','$phoneNum','$website','$faclityType','$capacity','$city','$country','$provinceID','$normalDayStartTime','$normalDayEndTime','$specialDayStartTime','$specialDayEndTime','$acceptWalkInAppointment');**

**Delete a Public Health Facility:**

**Delete from VaccinationFacility where facilityID = '$facilityID'**

**Edit a Public Health Facility:**

**update VaccinationFacility set facilityID= '$facilityID',name= '$name',address= '$address',phoneNum= '$phoneNum',website= '$website',faclityType= '$faclityType',capacity= '$capacity',city= '$city',country= '$country',provinceID= '$provinceID',normalDayStartTime= '$normalDayStartTime',normalDayEndTime= '$normalDayEndTime',specialDayStartTime= '$specialDayStartTime',specialDayEndTime= '$specialDayEndTime',acceptWalkInAppointment= '$acceptWalkInAppointment'**

**where VaccinationFacility.facilityID = '$oldFacilityID';**

**Display a Public Health Facility:**

**select \* from VaccinationFacility where VaccinationFacility.facilityID = '$id'**

**4.**

**// create**

**insert INTO vaccination(typeName,approvedDate,status,dateOfSuspension)**

**values ('$typeName','$approvedDate','$status','$dateOfSuspension');**

**// edit**

**update vaccination set typeName = '$typeName',approvedDate = '$approvedDate',status = '$status',dateOfSuspension = '$dateOfSuspension'**

**where vaccination.typeName = '$oldTypeName';**

**// delete**

**Delete from vaccination where typeName = '$typeName'**

**// display**

**select \* from vaccination**

**5.**

**// create**

**insert INTO infectedtype(infectedTypeID,infectedTypeName)**

**values ('$infectedTypeID','$infectedTypeName');**

**// edit**

**update infectedtype set infectedTypeID = '$infectionTypeID',infectedTypeName = '$infectionTypeName'**

**where infectedtype.infectedTypeID = '$oldinfectionTypeID'**

**//delete**

**Delete from infectedtype where infectedTypeID = '$typeID'**

**// dispaly**

**select \* from infectedtype**

**6.**

**// create**

**insert INTO groupage(ageGroup,min,max)**

**values ('$ageGroup','$min',$max);**

**// edit**

**update groupage set ageGroup = '$ageGroup',min = '$min',max = '$max'**

**where groupage.ageGroup = '$oldGroupID';**

**//delete**

**Delete from groupage where ageGroup = '$ageGroup'**

**// dispaly**

**select \* from groupage**

**7.**

**// create**

**insert INTO province(provinceID,provinceName,allowedAgeGroup)**

**values ('$provinceID','$provinceName',$allowedAgeGroup)**

**// edit**

**update province set provinceID = '$provinceID', provinceName = '$provinceName',allowedAgeGroup = '$ageGroup'**

**where province.provinceID = '$oldID'**

**//delete**

**Delete from province where provinceID = '$provinceID'**

**// dispaly**

**select \* from province**

**8.**

**// create**

**insert INTO province(provinceID,provinceName,allowedAgeGroup)**

**values ('$provinceID','$provinceName',$allowedAgeGroup)**

**// edit**

**update province set provinceID = '$provinceID', provinceName = '$provinceName',allowedAgeGroup = '$ageGroup'**

**where province.provinceID = '$oldID'**

**//delete**

**Delete from province where provinceID = '$provinceID'**

**// dispaly**

**select \* from province**

**9.**

**//create**

select AppointmentInformation.$facilityID, AppointmentInformation.$personID, AppointmentInformation.$vaccinationType, AppointmentInformation.$date, AppointmentInformation.$time, AppointmentInformation.$timeTo

from AppointmentInformation, Person, Province, VaccinationFacility

where AppointmentInformation.$personID = Person.personID and Person.ageGroup = Province.allowedAgeGroup and Province.provinceID = VaccinationFacility.provinceID and VaccinationFacility.facilityID = AppointmentInformation.facilityID";

//edit

update AppointmentInformation set facilityID = $facilityID, personID = $personID, vaccinationType = $vaccinationType, date = $date, time = $time, timeTo = $timeTo

where personID = $personID

//delete

Delete from appointment where AppointmentInformation.personID = '$personID'

//display

select \* from AppointmentInformation

**10.**

**//create**

SELECT Person.$personID,VaccinationFacility.$facilityName,PublicHealthWorker.$facilityJob, PublicHealthWorker.$startDate, PublicHealthWorker.$endtDate

FROM Person, PublicHealthWorker, VaccinationFacility

WHERE Person.$personID = PublicHealthWorker.personID AND PublicHealthWorker.facilityID = VaccinationFacility.facilityID";

**//edit**

update Person set personID = $personID

update VaccinationFacility set name = $facilityName

update PublicHealthWorker set jobTitle = $jobTitle, startWorkDate = $startWorkDate, endWorkDate = $endWorkDate

**//delete**

Delete from appointment where AppointmentInformation.personID = '$personID'

**//display**

SELECT Person.personID, VaccinationFacility.`name`,PublicHealthWorker.jobTitle, PublicHealthWorker.startWorkDate, PublicHealthWorker.endWorkDate

FROM Person, PublicHealthWorker, VaccinationFacility

WHERE Person.personID = PublicHealthWorker.personID AND PublicHealthWorker.facilityID = VaccinationFacility.facilityID

**11.**

SELECT AppointmentInformation.facilityID, AppointmentInformation.personID, AppointmentInformation.vaccinationType, AppointmentInformation.date, AppointmentInformation.time, AppointmentInformation.timeTo,Person.firstName,Avaliablity.avaliableSpot

FROM AppointmentInformation, Avaliablity, PublicHealthWorker, VaccinationFacility, Person, HealthWorkerFacilityID

WHERE AppointmentInformation.facilityID = Avaliablity.facilityID AND AppointmentInformation.time = Avaliablity.avaliableTimeFrom AND AppointmentInformation.timeTo = Avaliablity.avaliableTimeTo AND Avaliablity.facilityWorkerID = HealthWorkerFacilityID.facilityWorkerID AND HealthWorkerFacilityID.personID = PublicHealthWorker.personID AND PublicHealthWorker.personID = Person.personID

12.

SELECT Avaliablity.avaliableSpot, Avaliablity.avaliableTimeFrom, Avaliablity.date, Province.provinceName AS province\_name, VaccinationFacility.`name` AS facility\_name

FROM Avaliablity, Province, VaccinationFacility

WHERE Avaliablity.facilityID = VaccinationFacility.facilityID AND VaccinationFacility.provinceID = Province.provinceID

**13.**

**SELECT PublicHealthWorker.facilityID,Person.firstName,Person.lastName,Person.email,PublicHealthWorker.salary**

**FROM PublicHealthWorker,Person,VaccinationFacility**

**WHERE PublicHealthWorker.jobTitle = 'nurse'**

**AND PublicHealthWorker.personID = Person.personID**

**AND VaccinationFacility.facilityID = PublicHealthWorker.facilityID**

**AND VaccinationFacility.name = '$name'**

**AND PublicHealthWorker.startWorkDate> $date**

**AND PublicHealthWorker.endWorkDate< $date**

**ORDER BY PublicHealthWorker.salary DESC;**

**14.**

**SELECT VaccinationFacility.name,VaccinationFacility.address,**

**VaccinationFacility.phoneNum,VaccinationFacility.capacity,**

**timediff(VaccinationFacility.normalDayStartTime,VaccinationFacility.normalDayENdTime) as operatingHours**

**FROM VaccinationFacility left join**

**(SELECT VaccinationFacility.name,COUNT(\*)**

**FROM VaccinationFacility,PublicHealthWorker,AppointmentInformation,Person**

**WHERE VaccinationFacility.facilityID = AppointmentInformation.facilityID**

**AND AppointmentInformation.personID = Person.personID**

**AND Person.personID = PublicHealthWorker.personID**

**AND PublicHealthWorker.jobTitle = 'nurse'**

**AND PublicHealthWorker.startWorkDate> $date**

**AND PublicHealthWorker.endWorkDate< $date**

**GROUP BY VaccinationFacility.name) b ON VaccinationFacility.name!=b.name**

**GROUP BY VaccinationFacility.name;**

**15.**

**SELECT Person.firstName,Person.lastName,PublicHealthWorker.jobTitle,PublicHealthWorker.startWorkDate,PublicHealthWorker.endWorkDate**

**FROM VaccinationFacility,AppointmentInformation,Person,PublicHealthWorker**

**WHERE VaccinationFacility.facilityID = AppointmentInformation.facilityID**

**AND AppointmentInformation.personID = Person.personID**

**AND PublicHealthWorker.personID = Person.personID**

**AND VaccinationFacility.name = '$name'**

**AND PublicHealthWorker.startWorkDate> $date**

**AND PublicHealthWorker.endWorkDate< $date**

**;"**

**16.**

**SELECT Person.firstName,Person.lastName,Person.middleInitial,Person.city**

**FROM AppointmentInformation,Person**

**WHERE AppointmentInformation.personID = Person.personID**

**17.**

**SELECT DISTINCT a.personID,a.firstName, a.lastName, a.middleInitial, a.city**

**FROM (SELECT Person.personID,Person.firstName,Person.lastName,Person.middleInitial,Person.city**

**FROM VaccinatedHistory,Person**

**WHERE VaccinatedHistory.personID = Person.personID ) a,**

**(SELECT Person.personID,Person.firstName,Person.lastName,Person.middleInitial,Person.city**

**FROM AppointmentInformation,Person**

**WHERE AppointmentInformation.personID = Person.personID) b**

**WHERE a.personID = b.personID;**

**18.**

**SELECT Person.personID,Person.firstName,Person.lastName,Person.phoneNumber,COUNT(VaccinatedHistory.doseNum)as doseNum**

**FROM PublicHealthWorker,VaccinatedHistory,Person**

**WHERE PublicHealthWorker.personID = VaccinatedHistory.personID**

**AND PublicHealthWorker.personID = Person.personID**

**AND VaccinatedHistory.personID = Person.personID**

**GROUP BY Person.firstName**

**ORDER BY COUNT(VaccinatedHistory.doseNum) DESC;**

**19.**

**SELECT vaccinationfacility.name,vaccinationfacility.city,vaccinationfacility.address,vaccinationfacility.faclityType,vaccinationfacility.phoneNum,vaccinationfacility.capacity,**

**emp.total\_workers as total\_emp ,**

**does.total\_does as total\_Received\_Does,**

**bookings.total\_bookings as total\_bookings**

**FROM vaccinationfacility**

**LEFT JOIN (SELECT publichealthworker.facilityID, COUNT( distinct publichealthworker.personID) AS total\_workers**

**FROM publichealthworker**

**group by publichealthworker.facilityID) as emp**

**ON emp.facilityID = vaccinationfacility.facilityID**

**LEFT JOIN (select vaccinatedhistory.facilityID, COUNT(vaccinatedhistory.doseNum) as total\_does**

**FROM vaccinatedhistory**

**group by vaccinatedhistory.facilityID) as does**

**ON does.facilityID = vaccinationfacility.facilityID**

**LEFT JOIN (SELECT appointmentinformation.facilityID,COUNT(appointmentinformation.date) as total\_bookings**

**FROM appointmentinformation**

**WHERE appointmentinformation.date > '$today'**

**group by appointmentinformation.facilityID) as bookings**

**on bookings.facilityID = vaccinationfacility.facilityID**

**WHERE vaccinationfacility.city = 'Montreal'";**

**20.select person.firstName,person.lastName,person.dateOfBirth, VaccinationFacility.name,vaccinationFacility.address, appointmentinformation.date,appointmentinformation.time**

**from VaccinationFacility,appointmentinformation,person**

**where appointmentinformation.personID = $personID and appointmentinformation.facilityID = VaccinationFacility.facilityID and**

**person.personID = $personID;**

**select \* from infectedhistory**

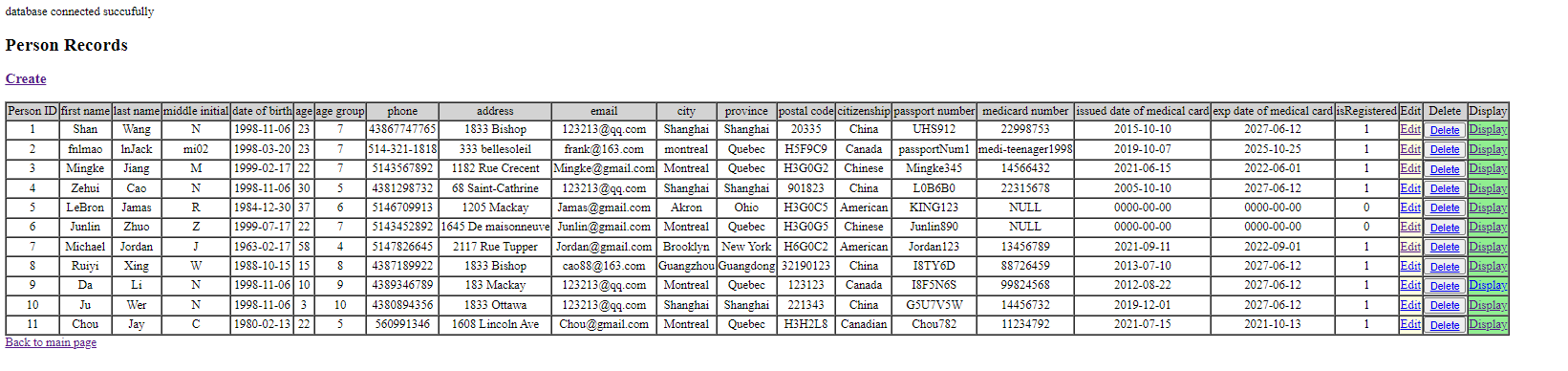
**where infectedhistory.personID = $personID;**

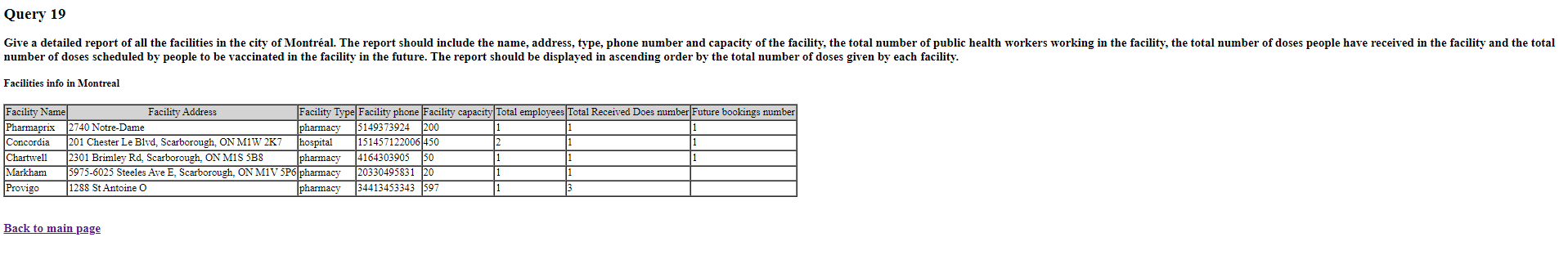
**select \***

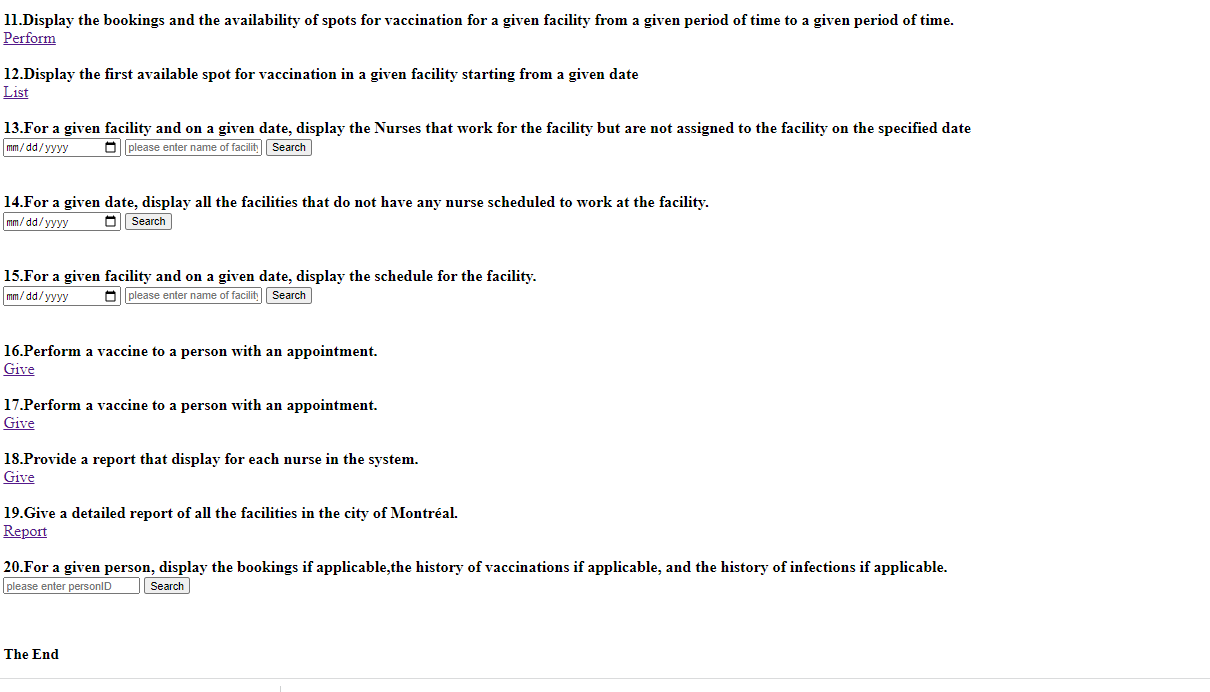
**from vaccinatedhistory**

**where vaccinatedhistory.personID = $personID ;**

**4.interface snapshots**

****

****

****