

CPSC 304 Project Cover Page

Milestone #: 4

Date: Nov.25th

Group Number: 68

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Yitan Li	40581902	m8q4m	liyitan2002@163.com
Shaoyun Tong	56409303	a4g9o	tshyun0224@gmail.com
Abby Hong	99385726	l8y1u	abbyhong0811@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Repository Link:

https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_a4g9o_l8y1u_m8q4m.git

SQL script to create all the tables and data in the database:

We include this file in our github repository.

Project Description:

A short description of the final project, and what it accomplished.

The final project is a web platform based application with a relational database as main support for organ transplant organizations to manage logistics and administration tasks. It contains a user interface that provides basic and useful operations which administrators in organ transplant organizations may use, such as adding waiting patients, updating donor information. It also enables queries from users, such as matching suitable waiting patients for an organ, and finding some experienced partnered hospitals. All these operations and queries access and manipulate the relational database we build for this application using sql and oracle database management.

Project Description:

A description of how your final schema differed from the schema you turned in. If the final schema differed, explain why.

We changed the birthday attribute of donors and waiting patients to age. So the domain changed to int in our final schema for this attribute, which is easy for us to write and execute more meaningful queries using aggregation with int data type. Also, we delete ON DELETE CASCADE for FK waitingTime in the table for WaitingTransplantation as suggestions, because we don't want the waiting time attribute deleted when it deletes from WaitingTimeMapUrgentDeg. We also changed hospitalID in the IndividualRecipientOperation table to have NOT NULL constraints.

University of British Columbia, Vancouver

Department of Computer Science

A copy of the schema and screenshots that show what data is present in each relation after the SQL script from item #2 is run.

Table: Organization

```
insert into Organization
values(1, 'Vancouver', '6088 walter gage road', 'BC Transplant center');

insert into Organization
values(2, 'Beijing', '8783 Beijing Road', 'Beijing Transplant center');

insert into Organization
values(3, 'Henan', '7384 WenHua road', 'Henan Transplant center');

insert into Organization
values(4, 'Shanghai', '7783 Nanjing West road', 'Shanghai Transplant center');

insert into Organization
values(5, 'Vancouver', '88 hollywood road', 'BC EyeBank');
```

ORG_ID	ORG_CITY	ORG_STREETADDR	ORG_NAME
1	Vancouver	6088 walter gage road	BC Transplant center
2	Beijing	8783 Beijing Road	Beijing Transplant center
3	Henan	7384 WenHua road	Henan Transplant center
4	Shanghai	7783 Nanjing West road	Shanghai Transplant center
5	Vancouver	88 hollywood road	BC EyeBank

Table: OrganTransplantOrganization

```
insert into OrganTransplantOrganization
values(2, 'China Organ research Foundation');

insert into OrganTransplantOrganization
values(1, 'BC Organ Transplant research Foundation');

insert into OrganTransplantOrganization
values(3, null);

insert into OrganTransplantOrganization
values(4, null);
```

```
SQL> SELECT * FROM OrganTransplantOrganization;
```

ORG_ID	TRANSRESEARCHFUND
2	China Organ research Foundation
1	BC Organ Transplant research Foundation
3	
4	

Table: CornealTransplantOrganization

```
insert into CornealTransplantOrganization
values(5, 'Jack B');
```

```
SQL> SELECT * FROM CornealTransplantOrganization;
```

ORG_ID	OPHTHADIRECTOR
5	Jack B

Table: Hospital

```
insert into Hospital
values(12, '5680 University Boulevard', 'UBC hospital');

insert into Hospital
values(65, '5738 Beijing Boulevard', 'Beijing hospital');

insert into Hospital
values(876, '1180 Zhengzhou Boulevard', 'Zhengzhou hospital');

insert into Hospital
values(234, '5680 shanghai Boulevard', 'Shanghai hospital');

insert into Hospital
values(132, '3380 No.3 Road', 'Richmond hospital');
```

```
SQL> SELECT * FROM Hospital;
```

HOSPITAL_ID	HOSPITALADDRESS	HOSPITALNAME
12	5680 University Boulevard	UBC hospital
65	5738 Beijing Boulevard	Beijing hospital
876	1180 Zhengzhou Boulevard	Zhengzhou hospital
234	5680 shanghai Boulevard	Shanghai hospital
132	3380 No.3 Road	Richmond hospital

Table: partnerWith

```
insert into partnerWith values(1, 12);
insert into partnerWith values(1, 132);
insert into partnerWith values(5, 132);
insert into partnerWith values(2, 65);
insert into partnerWith values(4, 234);
```

```
SQL> SELECT * FROM partnerWith;
```

ORG_ID	HOSPITAL_ID
1	12
1	132
2	65
4	234
5	132

Table: WaitingTimeMapUrgentDeg

```
insert into WaitingTimeMapUrgentDeg values(10000, 10);
insert into WaitingTimeMapUrgentDeg values(1, 1);
insert into WaitingTimeMapUrgentDeg values(8000, 8);
insert into WaitingTimeMapUrgentDeg values(9000, 9);
insert into WaitingTimeMapUrgentDeg values(3244, 4);
insert into WaitingTimeMapUrgentDeg values(0, 0);
```

```
SQL> SELECT * FROM WaitingTimeMapUrgentDeg;
```

WAITINGTIME	URGENTDEGREE
10000	10
1	1
8000	8
9000	9
3244	4
0	0

Table: WaitingTransplantation

```
insert into WaitingTransplantation values (1, 18, 'A', 1, 'liver', 'Sasa ');
insert into WaitingTransplantation values (2, 20, 'B', 10000, 'kidney', 'Kelly');
insert into WaitingTransplantation values (3, 42, 'O', 8000, 'corneal', 'Bibo');
insert into WaitingTransplantation values (4, 29, 'AB', 9000, 'lung', 'Kelly');
insert into WaitingTransplantation values (5, 77, 'O', 3244, 'corneal', 'Cici');
```

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
1	18	A	1	liver	Sasa
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici

Table: Manage

```
insert into Manage values(1, 1);
insert into Manage values(2, 2);
insert into Manage values(2, 3);
insert into Manage values(3, 3);
insert into Manage values(4, 4);
insert into Manage values(5, 5);
```

```
SQL> SELECT * FROM Manage;
```

ORG_ID	TRANSPLANTATIONID
1	1
2	2
2	3
3	3
4	4
5	5

Table: Donor

```
insert into Donor values(12345, 'A', 'alive', 123767867, 'Lily', 35, 'V6T1Z1');
insert into Donor values(23456, 'B', 'alive', 328247837, 'Coco', 21, 'V6T1Z2');
insert into Donor values(34567, 'O', 'alive', 23454545, 'Anna', 56, 'V8Z1Z1');
insert into Donor values(45678, 'AB', 'pass away', 123723445, 'Someone', null, 'V6T1Z1');
insert into Donor values(56789, 'RHAB', 'pass away', 1321586, 'Rick', 61, 'V6T1Z1');
```

```
SQL> SELECT * FROM Donor;
```

DONORPHN	BLOODTYPE	DONORSTATUS	DONORPHONE	DONORNAME	DONORAGE	ADDRESS
12345	A	alive	123767867	Lily	35	V6T1Z1
23456	B	alive	328247837	Coco	21	V6T1Z2
34567	O	alive	23454545	Anna	56	V8Z1Z1
45678	AB	pass away	123723445	Someone		V6T1Z1
56789	RHAB	pass away	1321586	Rick	61	V6T1Z1

Table: RegisterIn

```
insert into RegisterIn values(4, 12345, 'organ', '2022-08-01');
insert into RegisterIn values(2, 23456, 'organ', '2013-08-01');
insert into RegisterIn values(1, 34567, 'corneal', '2002-08-01');
insert into RegisterIn values(1, 45678, 'organ', '2012-09-21');
insert into RegisterIn values(5, 56789, 'corneal', '2019-01-01');
insert into RegisterIn values(4, 56789, 'organ', '2019-01-01');
insert into RegisterIn values(3, 12345, 'corneal', '2019-01-01');
```

```
SQL> SELECT * FROM RegisterIn;
```

ORG_ID	DONORPHN	REGISTRATI	REGISTERDATE
4	12345	organ	2022-08-01
2	23456	organ	2013-08-01
1	34567	corneal	2002-08-01
1	45678	organ	2012-09-21
5	56789	corneal	2019-01-01
4	56789	organ	2019-01-01
3	12345	corneal	2019-01-01

Table: DonorFamilyContactPerson

```
insert into DonorFamilyContactPerson values(12, 24254767, '143345@gmail.com', 'Kitty');
insert into DonorFamilyContactPerson values(34, 32652345, '245563@gmail.com', 'Jack');
insert into DonorFamilyContactPerson values(56, 2368754785, 'betty63@gmail.com', 'Betty');
insert into DonorFamilyContactPerson values(78, 77789654, 'kate63@gmail.com', 'Kate');
insert into DonorFamilyContactPerson values(90, 310876754, 'coco63@gmail.com', 'Coco');
```

```
SQL> SELECT * FROM DonorFamilyContactPerson;
```

CONTACTPHN	CONTACTPHONE	EMAIL	CONTACTNAME
12	24254767	143345@gmail.com	Kitty
34	32652345	245563@gmail.com	Jack
56	2368754785	betty63@gmail.com	Betty
78	77789654	kate63@gmail.com	Kate
90	310876754	coco63@gmail.com	Coco

Table: Has

```
insert into Has values(12, 12345);
insert into Has values(34, 23456);
insert into Has values(56, 34567);
insert into Has values(78, 45678);
insert into Has values(90, 56789);
insert into Has values(90, 45678);
```

```
SQL> SELECT * FROM Has;
```

CONTACTPHN	DONORPHN
12	12345
34	23456
56	34567
78	45678
90	45678
90	56789

Table: OrganTypeInfo

```
insert into OrganTypeInfo values('heart', 30, 1);
insert into OrganTypeInfo values('lung', 60, 2);
insert into OrganTypeInfo values('kidney', 120, 2);
insert into OrganTypeInfo values('corneal', 100, 5);
insert into OrganTypeInfo values('liver', 35, 1);
```

```
SQL> SELECT * FROM OrganTypeInfo;
```

ORGANTYPE	ORGANSURIVETIME	NUMTRANSTO
heart	30	1
lung	60	2
kidney	120	2
corneal	100	5
liver	35	1

Table: DonateOrgan

```
insert into DonateOrgan values(1, 'available', 'heart', 12345);
insert into DonateOrgan values(6, 'available', 'lung', 12345);
insert into DonateOrgan values(7, 'available', 'kidney', 12345);
insert into DonateOrgan values(8, 'available', 'corneal', 12345);
insert into DonateOrgan values(9, 'available', 'liver', 12345);
insert into DonateOrgan values(2, 'unavailable', 'corneal', 23456);
insert into DonateOrgan values(3, 'available', 'kidney', 34567);
insert into DonateOrgan values(4, 'unavailable', 'lung', 45678);
insert into DonateOrgan values(5, 'available', 'kidney', 56789);
```

```
SQL> SELECT * FROM DonateOrgan;
```

ORGANID	ORGANSTATUS	ORGANTYPE	DONORPHN
1	available	heart	12345
6	available	lung	12345
7	available	kidney	12345
8	available	corneal	12345
9	available	liver	12345
2	unavailable	corneal	23456
3	available	kidney	34567
4	unavailable	lung	45678
5	available	kidney	56789

Table: Recipient

```
insert into Recipient values(1, 'kaka', 1233435);
insert into Recipient values(2, 'Bibo', 234545);
insert into Recipient values(3, 'Lala', 778735);
insert into Recipient values(4, 'Cici', 446574);
insert into Recipient values(5, 'Papi', 130655);
insert into Recipient values(678, 'UBC life lab', 25634589);
insert into Recipient values(778, 'SFU life lab', 25653989);
insert into Recipient values(878, 'Uvic life lab', 1253989);
insert into Recipient values(978, 'UBC med lab', 224653989);
insert into Recipient values(378, 'UT life lab', 276546989);
insert into Recipient values(6, 'kiki', 633435);
insert into Recipient values(7, 'koko', 633445);
```

```
SQL> SELECT * FROM Recipient;
```

RECIPIENTID	NAME	PHONE
1	kaka	1233435
2	Bibo	234545
3	Lala	778735
4	Cici	446574
5	Papi	130655
678	UBC life lab	25634589
778	SFU life lab	25653989
878	Uvic life lab	1253989
978	UBC med lab	224653989
378	UT life lab	276546989
6	kiki	633435
7	koko	633445

Table: IndividualRecipientOperation

```
insert into IndividualRecipientOperation values(1, 'kaka', 1233435, 234, 12);
insert into IndividualRecipientOperation values(6, 'kiki', 633435, 235, 12);
insert into IndividualRecipientOperation values(7, 'koko', 633445, 245, 12);
insert into IndividualRecipientOperation values(2, 'Bibo', 234545, 334, 65);
insert into IndividualRecipientOperation values(3, 'Lala', 778735, 634, 876);
insert into IndividualRecipientOperation values(4, 'Cici', 446574, 934, 234);
insert into IndividualRecipientOperation values(5, 'Papi', 130655, 134, 132);
```


University of British Columbia, Vancouver
Department of Computer Science

```
SQL> SELECT * FROM IndividualRecipientOperation;
```

RECIPIENTID	RECIPIENTNAME	RECIPIENTPHONE	INDIVIDUALPHN	HOSPITAL_ID
1	kaka	1233435	234	12
6	kiki	633435	235	12
7	koko	633445	245	12
2	Bibo	234545	334	65
3	Lala	778735	634	876
4	Cici	446574	934	234
5	Papi	130655	134	132

Table: LaboratoryRecipient

```
insert into LaboratoryRecipient values(678, 'UBC life lab', '6798 universityroad', 25634589);
insert into LaboratoryRecipient values(778, 'SFU life lab', '8988 SFU road', 25653989);
insert into LaboratoryRecipient values(878, 'Uvic life lab', '6238 Victoriaroad', 1253989);
insert into LaboratoryRecipient values(978, 'UBC med lab', '6799 university road', 224653989);
insert into LaboratoryRecipient values(378, 'UT life lab', '6798 Torontoroad', 276546989);
```

```
SQL> SELECT * FROM LaboratoryRecipient;
```

RECIPIENTID	RECIPIENTNAME	LABADDRESS	RECIPIENTPHONE
678	UBC life lab	6798 universityroad	25634589
778	SFU life lab	8988 SFU road	25653989
878	Uvic life lab	6238 Victoriaroad	1253989
978	UBC med lab	6799 university road	224653989
378	UT life lab	6798 Torontoroad	276546989

Table: Accept

```
insert into Accept values('2022-09-06', 1, 1);
insert into Accept values('2021-08-06', 2, 2);
insert into Accept values('2020-07-06', 3, 678);
insert into Accept values('2010-09-16', 4, 878);
insert into Accept values('2018-09-26', 5, 978);
```

```
SQL> SELECT * FROM Accept;
```

ACCEPTDATE	ORGANID	RECIPIENTID
2022-09-06	1	1
2021-08-06	2	2
2020-07-06	3	678
2010-09-16	4	878
2018-09-26	5	978

A list of all SQL queries used:

“\$” represents the value from the web input

1. Insert Query

In php file, we called executeBoundSQL() to insert waiting transplantation patient while the real sql statement is:

```
Insert into WaitingTransplantation  
values($transplantationID,$patientBirthday, $patientBloodType  
, $waitingTime, $neededOrgan, $patientName)
```

2. Delete Query

```
DELETE FROM WaitingTransplantation  
WHERE transplantationID = '$patientID'
```

3. Update Query

```
UPDATE Donor  
SET donorStatus = '$donor_newstatus'  
WHERE donorPhn = '$donor_phn'
```

4. Selection Query

```
SELECT WT.transplantationID, WT.patientName, WT.patientBloodType,  
WT.neededOrgan, WT.waitingTime  
FROM WaitingTransplantation WT  
WHERE WT.neededOrgan = '$neededOrgan'  
ORDER BY WT.waitingTime DESC
```

5. Projection Query

```
$choice = customer selected from web  
SELECT $choice, donorName  
From Donor, DonateOrgan  
WHERE Donor.donorPhn = DonateOrgan.donorPhn
```

6. Join Query

```
$selected = DonorFamilyContactPerson.email or  
DonorFamilyContactPerson.contactName  
or DonorFamilyContactPerson.contactPhone  
or DonorFamilyContactPerson.contactPhn  
  
SELECT $selected FROM Donor, Has, DonorFamilyContactPerson
```

```
WHERE Donor.donorPhn = '$donorPhn' AND Donor.donorPhn = Has.donorPhn AND  
Has.contactPhn = DonorFamilyContactPerson.contactPhn
```

7. Aggregation with Group by Query

```
SELECT MAX(WaitingTransplantation.waitingTime),  
WaitingTransplantation.neededOrgan  
FROM WaitingTransplantation  
GROUP BY WaitingTransplantation.neededOrgan
```

8. Aggregation with Having Query

```
SELECT Hospital.hospitalName From Hospital, IndividualRecipientOperation  
WHERE Hospital.hospital_id = IndividualRecipientOperation.hospital_id  
GROUP BY Hospital.hospitalName  
HAVING COUNT(IndividualRecipientOperation.recipientId) > 1
```

9. Nested Aggregation with Group By

```
SELECT organType, count(*)  
FROM DonateOrgan  
GROUP BY organType  
HAVING count(*) <= all (SELECT count(*)  
FROM DonateOrgan D  
GROUP BY D.organType)
```

10. Division Query

```
SELECT Distinct D.donorPhn, D.donorName  
FROM Donor D  
WHERE NOT EXISTS(  
    (SELECT OTI.organType FROM OrganTypeInfo OTI)  
    MINUS  
    (SELECT DO.organType  
    FROM DonateOrgan DO  
    WHERE DO.donorPhn = D.donorPhn))
```

Screenshots of the sample output of the queries using the GUI.

1. Insert Query

website looks like

Add a new patient to the waiting list (Insert)

Name: Organ Needed:

Age: Blood Type:

before we insert

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
1	18	A	1	liver	Sasa
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici

after we insert

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
1	18	A	1	liver	Sasa
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici
1.7505E+15	29	AB	0	heart	tsy

2. Delete Query

website looks like

Delete A Waiting Patient (Delete)

Waiting Patient ID:

before we delete

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
1	18	A	1	liver	Sasa
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici
1.7505E+15	29	AB	0	heart	tsy

after we delete

University of British Columbia, Vancouver

Department of Computer Science

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici
1.7505E+15	29	AB	0	Heart	tsy

3. Update Query

website looks like

Update Donor Information (Update)

Donor phn:

Status:

before we update

```
SQL> SELECT * FROM Donor;
```

DONORPHN	BLOODTYPE	DONORSTATUS	DONORPHONE	DONORNAME	DONORAGE	ADDRESS
12345	A	alive	123767867	Lily	35	V6T1Z1
23456	B	alive	328247837	Coco	21	V6T1Z2
34567	O	alive	23454545	Anna	56	V8Z1Z1
45678	AB	pass away	123723445	Someone		V6T1Z1
56789	RHAB	pass away	1321586	Rick	61	V6T1Z1

after we update

```
SQL> SELECT * FROM Donor;
```

DONORPHN	BLOODTYPE	DONORSTATUS	DONORPHONE	DONORNAME	DONORAGE	ADDRESS
12345	A	pass away	123767867	Lily	35	V6T1Z1
23456	B	alive	328247837	Coco	21	V6T1Z2
34567	O	alive	23454545	Anna	56	V8Z1Z1
45678	AB	pass away	123723445	Someone		V6T1Z1
56789	RHAB	pass away	1321586	Rick	61	V6T1Z1

4. Selection Query

website looks like

Match Suitable Waiting Patient

Filter Waiting Patients By Needed Organ (Selection)

Needed Organ:

the waiting patients table is

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici
1.7505E+15	29	AB	0	Heart	tsy

output

University of British Columbia, Vancouver

Department of Computer Science

Selection Query

Patient ID Patient Name Blood Type Needed Organ

3	Bibo	O	corneal
5	Cici	O	corneal

5. Projection Query

website looks like

View donated Organ/Status/Blood Type of all donors (Projection)

Select the attributes of donors

View

Donor Organ
✓ Donor Status
Donor Blood

the donor table is

```
SQL> SELECT * FROM Donor;
```

DONORPHN	BLOODTYPE	DONORSTATUS	DONORPHONE	DONORNAME	DONORAGE	ADDRESS
12345	A	pass away	123767867	Lily	35	V6T1Z1
23456	B	alive	328247837	Coco	21	V6T1Z2
34567	O	alive	23454545	Anna	56	V8Z1Z1
45678	AB	pass away	123723445	Someone		V6T1Z1
56789	RHAB	pass away	1321586	Rick	61	V6T1Z1

output

donorStatus Donor Name

pass away	Lily
alive	Coco
alive	Anna
pass away	Someone
pass away	Rick

6. Join Query

website looks like

Show Patient Family Information (Join)

Donor #phn: 23456

Select Donar Family Attribute

Show Patient Family Information

Family Email
✓ Family Name
Family Phone
Family Phn

the donor table please see above, the donor family table is

```
SQL> SELECT * FROM DonorFamilyContactPerson;
```

CONTACTPHN	CONTACTPHONE	EMAIL	CONTACTNAME
12	24254767	143345@gmail.com	Kitty
34	32652345	245563@gmail.com	Jack
56	2368754785	betty63@gmail.com	Betty
78	77789654	kate63@gmail.com	Kate
90	310876754	coco63@gmail.com	Coco

output

Join Query For Donar Phn Is #23456

Family Contact Name

Jack

7. Aggregation with Group By Query

website looks like

Show the Max Waiting Time for Each Organ (Group By)

View

the waitingTransplantation table is

```
SQL> SELECT * FROM WaitingTransplantation;
```

TRANSPLANTATIONID	PATIENTAGE	PATIENTBLO	WAITINGTIME	NEEDEDORGAN	PATIENTNAME
1	18	A	1	liver	Sasa
2	20	B	10000	kidney	Kelly
3	42	O	8000	corneal	Bibo
4	29	AB	9000	lung	Kelly
5	77	O	3244	corneal	Cici
1.7505E+15	29	AB	0	heart	tsy

output

Group By Query

Max Waiting Time Need Organ

8000	corneal
1	liver
10000	kidney
9000	lung
0	heart

8. Aggregation with Having Query

website looks like

Find some experienced hospitals

at least two recipient operated in (Having)

Find hospitals

the Hospital table is

```
SQL> SELECT * FROM Hospital;
```

HOSPITAL_ID	HOSPITALADDRESS	HOSPITALNAME
12	5680 University Boulevard	UBC hospital
65	5738 Beijing Boulevard	Beijing hospital
876	1180 Zhengzhou Boulevard	Zhengzhou hospital
234	5680 shanghai Boulevard	Shanghai hospital
132	3380 No.3 Road	Richmond hospital

the IndividualRecipientOperation table is

```
SQL> SELECT * FROM IndividualRecipientOperation;
```

RECIPIENTID	RECIPIENTNAME	RECIPIENTPHONE	INDIVIDUALPHN	HOSPITAL_ID
1	kaka	1233435	234	12
6	kiki	633435	235	12
7	koko	633445	245	12
2	Bibo	234545	334	65
3	Lala	778735	634	876
4	Cici	446574	934	234
5	Papi	130655	134	132

output

Name

UBC hospital

9. Nested Aggregation with Group By Query

website looks like

Find the most shortage organ type

Find the type of organ with least number of donors registered to donate (Nested aggregation)

View

the DonateOrgan table is

```
SQL> SELECT * FROM DonateOrgan;
```

ORGANID	ORGANSTATUS	ORGANTYPE	DONORPHN
1	available	heart	12345
6	available	lung	12345
7	available	kidney	12345
8	available	corneal	12345
9	available	liver	12345
2	unavailable	corneal	23456
3	available	kidney	34567
4	unavailable	lung	45678
5	available	kidney	56789

output

Nested Aggregation With Group By

Organ Type Number of Registed Donor

liver 1

heart 1

10. Division Query

website looks like

Find Donors Who Donate All Types of Organ (Division)

View

the DonateOrgan table(see #9), the Donor table(see #5) , the organTypeInfo table

```
SQL> SELECT * FROM OrganTypeInfo;
```

ORGANTYPE	ORGANSURIVETIME	NUMTRANSTO
heart	30	1
lung	60	2
kidney	120	2
corneal	100	5
liver	35	1

output

Division Query

Donor PHN Donor Name

12345 Lily