

**A**  
**Project Activity Report**  
**Submitted for Database Management System**  
**(UCS-310)**

---

**TOPIC: BLOOD BANK DATABASE  
MANAGEMENT SYSTEM**

---

**BE Second Year, COE**

**Submitted To:**

**Ms. Tanya Garg**



DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING  
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,  
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB INDIA

Jan-June 2022

# **INDEX**

<b>Sr. No.</b>	<b>Contents</b>	<b>Page No.</b>
1	Problem Statement	<b>3</b>
2	ER Diagram	<b>4</b>
3	ER to Table	<b>8</b>
4	Normalization	<b>15</b>
5	SQL/PLSQL	<b>28</b>

## **Problem Statement**

Patients require blood to survive operations, cancer treatments, chronic diseases, and traumatic traumas. This lifesaving care begins with a thoughtful contribution from one individual. The demand for blood is never-ending.

In 2016, 10.9 million donations were recorded, whereas in 2020, 12.7 million donations were reported, little less than expected but still sufficient despite the epidemic. As engineers, we have attempted to make the process of maintaining all records easier.

We're building a database that will keep all of the information regarding donors, recipients, blood inventories, organisations, and camps which can be used to retrieve, update, and collect data.

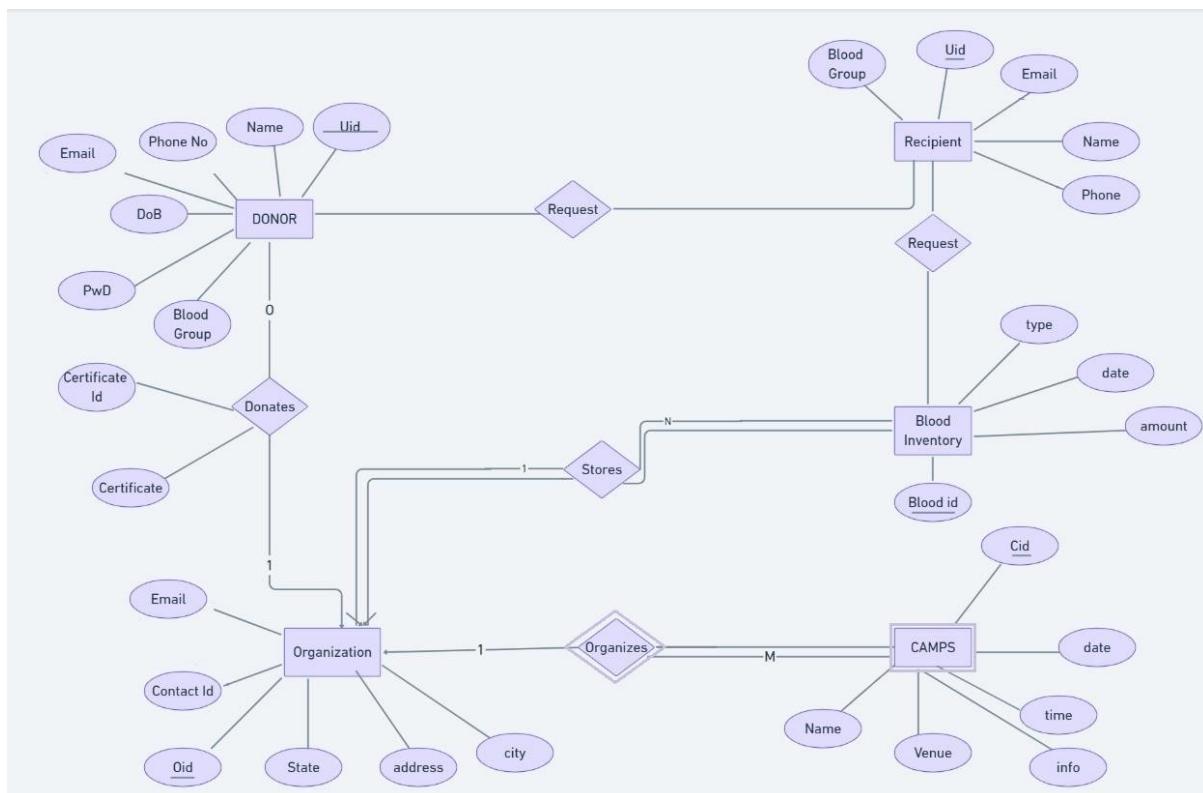
The main benefit of having a database for blood bank administration is the quick and easy retrieval of information. Manual procedures can be eliminated. Databases help you save time and enhance the quality and consistency of your information.

# ER Diagram

**ER Model** stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyse data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.

The ER Entity Relation diagram is a visual representation of all the entities and their attributes with their relationships.

The ER diagram made for our project consists of two entities with their 18 attributes describing how the various processes and entities are related in our database.



# ER-DIAGRAM DESCRIPTION

## ENTITIES AND ATTRIBUTES:

**Donor:** Donor is the physical entity or specifically the customer who would like to provide his details for blood donation which serves the motive for our database.

Attributes associated with this entity are:

UID: The UID is a unique **primary key** for the donor and will be a numeric value for every customer.

Name: It is a variable string field provided for the user to provide their name.

Phone no: It is a 10-digit phone number associated with the donor.

Email: It is a varchar2 type field to store email id of the user.

DOB: It is a **single value** attribute to store date of birth in dd-mm-yyyy format.

Blood Group: It is a **single valued** varchar2 type field used for storing the blood group of donor.

PwD: It is an attribute used to store the Password of the user in encrypted format.

Donor entity can be associated with '*donates*' and '*request' processes directly*.

**Organisation:** This is a physical entity related to donor and blood inventory directly by many to many relationship. The donor and organization both participate partially in this relation. The attributes of this entities are explained further.

Attributes of the entity mentioned are:

Email: It is a varchar2 type field to store email id.

State: It is a single valued field that stores a valid state name as a part of their address.

Address: It is a single valued field that stores local/ regional address of the user.

City: It is a single valued field that stores valid city name for the organization.

Contact ID: It is a field which identifies the donor and receiver individually.

OID: Organization ID a single valued key to uniquely identify an organization.

Donation: This is a relationship between two entities, ‘Donor’ and ‘Organisation’.

Its attributes are:

Certificate ID: This is a **unique** certificate ID when they donate blood.

Certificate: This refers to the certificate issued by the organisation for the global cause of donating blood.

Recipient: This is a physical entity which provides the details of the recipients of blood and is related to ‘donor’ entity by the relation ‘request’ and also ‘requests’ blood from the ‘blood inventory’ entity.

Attributes associated with this entity are:

UID: The UID is a **primary key** for the recipient and will be a numeric value for every customer.

Blood Group: It is an attribute which will tell about the blood group needed by the recipient and will be a string.

Email: It is a string storing the email addresses of the recipients.

Name: It is the attribute containing the names of all the recipients of string type.

Phone: It is an attribute storing the phone no. of recipients and is of type number.

Blood inventory: This is a physical entity related to ‘Organization’ entity by many to many relationships of ‘stores’ as it stores all the information about the blood donors and to ‘recipient’ by the relation ‘request’. The relationship between blood inventory and organisation is having total participation from both sides.

Attributes associated with this entity are:

Blood id: It is a **primary key** attribute which is a number and is given to each donor uniquely.

Type: It is a string attribute for storing the blood type stored.

Amount: It is a number attribute storing the price of blood.

Date: It is a single valued attribute of ‘date’ type.

**Camps:** This is a physical entity related to ‘Organization’ entity by many to one relationship of ‘stores’ as it stores all the information about the camps organized by various organisations. The relationship between camps and organisation is having total participation and partial participation respectively. It is a case of weak relationship with camps being the weak entity.

Attributes associated with this entity are

Cid: It is the primary key of weak entity named camps. It stores a numeric value that can uniquely identify each camp.

Name: It is a single valued attribute that can store the name of each camp in string format(varchar).

Date: It is a single valued attribute of ‘date’ type.

Time: It is a single valued attribute of ‘time’ type.

Venue: It denotes the location of the camp organized and stores it in varchar format.

Info: It stores some miscellaneous information about the camp in varchar format.

# ER to Table

## Creating tables

```
1  create database blood_bank;
2 * use blood_bank
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Action Output		
#	Time	Action
1	10:58:13	create database blood_bank
2	10:58:24	use blood_bank
3	10:59:15	drop database blood_bank
4	11:00:23	create database blood_bank
5	11:00:23	use blood_bank

```
create table donor( uid  
int, name varchar(20),  
phone_no bigint(10),  
email varchar(20), DOB  
date, pwd varchar(30),  
blood_group varchar(5),  
primary key (uid) );
```

Action Output		
#	Time	Action
1	10:58:24	use blood_bank
2	10:59:15	drop database blood_bank
3	11:00:23	create database blood_bank
4	11:00:23	use blood_bank
5	11:09:19	create table donor( uid int, name varchar(20), phone_no bigint(10), email varchar(20), DOB date, pwd varchar(30), blood_group varchar(5), primary key (uid) );

```
create table recipient(
```

```

uid int primary key,
name varchar(20),
phone_no bigint(10),
email varchar(20),
blood_group(5)
);

```

The screenshot shows the MySQL Workbench interface. In the top query editor (Query 1), the SQL code for creating the recipient table is entered. In the bottom output window, the execution log shows the following actions:

#	Time	Action	Message	Duration / Fetch
4	11:00:23	create database blood_bank	1 row(s) affected	0.031 sec
5	11:00:23	use blood_bank	0 row(s) affected	0.000 sec
6	11:09:19	create table donor(uid int, name varchar(20), phone_no bigint(10), email varchar(20), DOB date, pwd varchar(30)...)	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.157 sec
7	11:11:06	desc donor	7 row(s) returned	0.047 sec / 0.000 sec
8	11:14:51	create table recipient(uid int, name varchar(20), phone_no bigint(10), email varchar(20), blood_group varchar(5), p...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.062 sec

```

create table request(
uid1 int, uid2
int,
foreign key (uid1) references donor(uid),
foreign key (uid2) references
recipient(uid));

```

The screenshot shows the MySQL Workbench interface with a query editor and an output pane.

**Query Editor:**

```

1 • create table request(
2   uid1 int,
3   uid2 int,
4   foreign key (uid1) references donor(uid),
5   foreign key (uid2) references recipient(uid) );
6

```

**Output Pane:**

Action	Time	Message	Duration / Fetch
use blood_bank	5 11:00:23	0 row(s) affected	0.000 sec
create table donor	6 11:09:19	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.157 sec
desc donor	7 11:11:06	7 row(s) returned	0.047 sec / 0.000 sec
create table recipient	8 11:14:51	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.062 sec
create table request	9 11:21:03	0 row(s) affected	0.172 sec

```

create table organisation( oid
int,
contact_id bigint(10),
email varchar(20), state
varchar(20), address
varchar(50), city
varchar(20), primary
key (oid));

```

The screenshot shows the MySQL Workbench interface with a query editor and an output window.

**Query Editor:**

```

1 • create table organisation(
2   oid int,
3   contact_id bigint(10),
4   email varchar(20),
5   state varchar(20),
6   address varchar(50),
7   city varchar(20),
8   primary key (oid) );
9
10

```

**Output Window:**

Action	Time	Action	Message	Duration / Fetch
6	11:09:19	create table donor(uid int, name varchar(20), phone_no bigint(10), email varchar(20), DOB date, pwd varchar(30), ...)	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.157 sec
7	11:11:06	desc donor	7 row(s) returned	0.047 sec / 0.000 sec
8	11:14:51	create table recipient(uid int, uid2 int, foreign key (uid) references donor(uid), foreign key (uid2) references recipi...)	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.062 sec
9	11:21:03	create table request(uid1 int, uid2 int, foreign key (uid1) references donor(uid), foreign key (uid2) references recipi...)	0 row(s) affected	0.172 sec
10	11:27:12	create table organisation(oid int, contact_id bigint(10), email varchar(20), state varchar(20), address varchar(50), c...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.110 sec

```

create table donation(
certificate_id bigint(4),
uid int primary key,
name varchar(30),
phone_no bigint(10),
email varchar(20), DOB
date, pwd varchar(30),
blood_group
varchar(5),
oid1 int,
foreign key (oid1) references organisation(oid) );

```

The screenshot shows a SQL database interface with two main windows. The top window is titled 'Query 1' and contains the following SQL code:

```

1 • create table donation(
2   certificate_id bigint(4),
3   uid int primary key,
4   name varchar(30),
5   phone_no bigint(10),
6   email varchar(20),
7   DOB date,
8   pwd varchar(30),
9   blood_group varchar(5),
10  oid1 int,
11  foreign key (oid1) references organisation(oid) );
12
13

```

The bottom window is titled 'Output' and displays the execution log:

Action	Time	Message	Duration / Fetch
7	11:11:06	desc donor	0.047 sec / 0.000 sec
8	11:14:51	create table recipient((uid int, name varchar(20), phone_no bigint(10), email varchar(20), blood_group varchar(5), p... 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.062 sec
9	11:21:03	create table request((uid1 int, uid2 int, foreign key (uid1) references donor(uid), foreign key (uid2) references recipi... 0 row(s) affected	0.172 sec
10	11:27:12	create table organisation((oid int, contact_id bigint(10), email varchar(20), state varchar(20), address varchar(50), c... 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.110 sec
11	11:47:41	create table donation(certificate_id bigint(4), uid int primary key, name varchar(30), phone_no bigint(10), email var...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1... 0.125 sec

create table organizes(

oid1 int, cid int

primary key, date1

date, timer time, info

varchar(30), venue

varchar(30), name

varchar(30),

foreign key (oid1) references organisation(oid) );

Query 1 sys\_config employee employee employee student class class employee SQL File 12\*

```

1 • o create table organizes(
2   oid1 int,
3   cid int primary key,
4   date1 date,
5   timer time,
6   info varchar(30),
7   venue varchar(30),
8   name varchar(30),
9   foreign key (oid1) references organisation(oid) );
10
11

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

#	Time	Action	Message	Duration / Fetch
8	11:14:51	create table recipient( uid int, name varchar(20), phone_no bigint(10), email varchar(20), blood_group varchar(5), p...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.062 sec
9	11:21:03	create table request( uid1 int ,uid2 int ,foreign key (uid1) references donor(uid),foreign key (uid2) references recipi...	0 row(s) affected	0.172 sec
10	11:27:12	create table organisation( oid int , contact_id bigint(10), email varchar(20), state varchar(20), address varchar(50), c...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.110 sec
11	11:47:41	create table donation( certificate_id bigint(4), uid int primary key, name varchar(30),phone_no bigint(10),email var...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release, 1...	0.125 sec
12	12:02:09	create table organizes(oid1 int ,cid int primary key ,date1 date ,timer time ,info varchar(30) ,venue varchar(30) ,na...	0 row(s) affected	0.109 sec

Context Help Snippets

create table inventory( oid1

int,

bloodid int primary key,

btype varchar(5), date1

date, amount int

foreign key (oid1) references organisation(oid));

Query 1 sys\_config employee employee employee employee student class class employee SQL File 12\*

```

1 • o create table inventory(
2   oid1 int,
3   bloodid int primary key,
4   btype varchar(5),
5   date1 date,
6   amount int,
7   foreign key (oid1) references organisation(oid) );
8

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

#	Time	Action	Message	Duration / Fetch
10	11:27:12	create table organisation( oid int , contact_id bigint(10), email varchar(20), state varchar(20), address varchar(50), c...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.110 sec
11	11:47:41	create table donation( certificate_id bigint(4), uid int primary key, name varchar(30),phone_no bigint(10),email var...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release, 1...	0.125 sec
12	12:02:09	create table organizes(oid1 int ,cid int primary key ,date1 date ,timer time ,info varchar(30) ,venue varchar(30) ,na...	0 row(s) affected	0.109 sec
13	12:06:00	select time('19:20:10') LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14	12:14:31	create table inventory(oid1 int ,bloodid int primary key ,btype varchar(5), date1 date ,amount int ,foreign key (oid1)...	0 row(s) affected	0.141 sec

Context Help Snippets

```

create table blood(
    bloodid int primary key,
    btype varchar(5),
    date1 date, amount int
);

```

The screenshot shows a SQL query editor window. At the top, there are tabs for 'Query 1' and several system tables like 'sys\_config', 'employee', etc. Below the tabs is a toolbar with icons for file operations, search, and help. The main area contains the SQL code for creating the 'blood' table. To the right of the code, there is a message box stating: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' Below the code, there is an 'Output' section titled 'Action Output'. It lists several actions with their times, descriptions, messages, and durations. For example, action 11 at 11:47:41 creates the 'blood' table, and action 12 at 12:02:09 creates the 'organizes' table.

Action	Time	Message	Duration / Fetch
11 11:47:41	create table donation(certificate_id bigint(4), uid int primary key, name varchar(30), phone_no bigint(10), email var...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1...	0.125 sec
12 12:02:09	create table organizes(oid1 int, oid int primary key, date1 date, timer time, info varchar(30), venue varchar(30), na...	0 row(s) affected	0.109 sec
13 12:06:00	select time('19:20:10') LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14 12:14:31	create table inventory(oid1 int, bloodid int primary key, btype varchar(5), date1 date, amount int, foreign key (oid1)...	0 row(s) affected	0.141 sec
15 12:18:54	create table blood(bloodid int primary key, btype varchar(5), date1 date, amount int)	0 row(s) affected	0.140 sec

```

create table recipient1(
    bloodgroup varchar(5),
    uid int primary key,
    email varchar(30),
    name varchar(20),
    phone bigint(10), bid1
    int,
    foreign key(bid1) references blood(bloodid));

```

```

1 • create table recipient1(
2   bloodgroup varchar(5),
3   uid int primary key,
4   email varchar(30),
5   name varchar(20),
6   phone bigint(10),
7   bid1 int,
8   foreign key(bid1) references blood(bloodid));

```

Output

#	Time	Action	Message	Duration / Fetch
12	12:02:09	create table organizes(oid1 int, cid int primary key, date1 date, timer time, info varchar(30), venue varchar(30), na...	0 row(s) affected	0.109 sec
13	12:06:00	select time('19:20:10') LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14	12:14:31	create table inventory(oid1 int, bloodid int primary key, btype varchar(5), date1 date, amount int, foreign key (oid1)...)	0 row(s) affected	0.141 sec
15	12:18:54	create table blood(bloodid int primary key, btype varchar(5), date1 date, amount int )	0 row(s) affected	0.140 sec
16	12:26:23	create table recipient1(bloodgroup varchar(5), uid int primary key, email varchar(30), name varchar(20), phone big...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.125 sec

## Normalization

### 1NF/First Normal Form

A table is said to be in first normal form if there are no multiple values for a cell in a table. The tables used in our project such as DONOR, DONATIONS and so on are in first normal form as they do not accept multiple values of an attribute per tuple.

The fields such as Email, Phone number can have multiple values per tuple so either multiple columns such as phone1, phone2 were to be provided or decomposition was to be done which was preferred.

### 2NF/Second Normal Form

This normal form states that partial functional dependencies cannot exist. To keep tables in second normal form, primary keys assigned have only one attribute in case of each table.

We have primary keys such as UID, OID etc which consist of only one attribute which solves the problem of partial functional dependency.

### 3NF/ Third Normal Form

Third normal form states that transitive functional dependencies cannot exist.

The 5 entities in our ER diagram are decomposed into 9 tables including tables for relationships. This is done in order to separate attributes into multiple tables such that no transitive relationship exists.

## Inserting Values

```
insert into donor values(0001,'Ram',9898989898,'1@gmailcom','2001-01-01','xyzabc','b+');  
insert into donor values(0002,'Ama',9898989898,'2@gmailcom','2001-01-02','xyzabc','A+');  
insert into donor values(0003,'jili',9898989898,'3@gmailcom','2001-01-03','xyzabc','o+'); insert  
into donor values(0004,'Juli',9898989898,'4@gmailcom','2001-01-04','xyzabc','A+'); insert into  
donor values(0005,'aka',9898989898,'5@gmailcom','2001-01-05','xyzabc','B+'); insert into  
donor values(0006,'baka',9898989898,'6@gmailcom','2001-01-06','xyzabc','C+'); insert into  
donor values(0007,'daka',9898989898,'7@gmailcom','2001-01-07','xyzabc','A+'); insert into  
donor values(0008,'sid',9898989898,'8@gmailcom','2001-01-10','xyzabc','B+'); insert into  
donor values(0009,'mid',9898989898,'9@gmailcom','2001-01-11','xyzabc','C+'); insert into  
donor values(0010,'rewa',9898989898,'10@gmailcom','2001-01-12','xyzabc','B+'); insert into  
donor values(0011,'raywa',9898989898,'11@gmailcom','2001-01-13','xyzabc','AB+'); insert into  
donor values(0012,'rahwa',9898989898,'12@gmailcom','2001-01-14','xyzabc','O-'); insert into  
donor values(0013,'gourab',9898989898,'13@gmailcom','2001-01-15','xyzabc','A+'); insert into  
donor values(0014,'Gaurav',9898989898,'14@gmailcom','2001-01-16','xyzabc','b+'); insert into  
donor values(0015,'Kasis',9898989898,'15@gmailcom','2001-01-17','xyzabc','AB-'); insert into  
donor values(0016,'Kosis',9898989898,'16@gmailcom','2001-01-18','xyzabc','A-'); insert into  
donor values(0017,'kasol',9898989898,'17@gmailcom','2001-01-19','xyzabc','B+'); insert into  
donor values(0018,'Bani',9898989898,'18@gmailcom','2001-01-20','xyzabc','AB+'); insert into  
donor values(0019,'Banno',9898989898,'19@gmailcom','2001-01-21','xyzabc','B-'); insert into  
donor values(0021,'Rama',9898949898,'1@gmailcom','2001-01-01','xyzfbc','b+'); insert into  
donor values(0022,'Amaa',9898949898,'2@gmailcom','2001-01-02','xyzfbc','A+'); insert into  
donor values(0023,'jilia',9898289898,'3@gmailcom','2001-01-03','xyfabc','o+'); insert into  
donor values(0024,'Julia',9898589898,'4@gmailcom','2001-01-04','xyfabc','A+'); insert into  
donor values(0025,'akaa',9898969898,'5@gmailcom','2001-01-05','xyfabc','B+'); insert into  
donor values(0026,'bakaa',9898589898,'6@gmailcom','2001-01-06','xfzabc','C+'); insert into  
donor values(0027,'dakaa',9898889898,'7@gmailcom','2001-01-07','xyzafc','A+'); insert into
```

```

donor values(0028,'sida',9898979898,'8@gmailcom','2001-01-10','xyzabf','B+'); insert into
donor values(0029,'mida',9898959898,'9@gmailcom','2001-01-11','xyzafc','C+'); insert into
donor values(0020,'rewaa',9898389898,'10@gmailcom','2001-01-12','xyzfbc','B+'); insert into
donor values(0021,'raywaa',9891989898,'11@gmailcom','2001-01-13','xyfabc','AB+'); insert
into donor values(0022,'rahwaa',9891989898,'12@gmailcom','2001-01-14','xyfabc','O-'); insert
into donor values(0023,'gouraba',9838989898,'13@gmailcom','2001-01-15','xyzbbc','A+');
insert into donor values(0024,'Gaurava',9848989898,'14@gmailcom','2001-01-
16','xyzabc','B+'); insert into donor values(0025,'Kasisa',9898589898,'15@gmailcom','2001-01-
17','xyzabv','AB-'); insert into donor values(0026,'Kosisa',9898589898,'16@gmailcom','2001-01-
18','xyzabn','A-'); insert into donor values(0027,'kasola',9898589898,'17@gmailcom','2001-01-
19','xyzabn','B+'); insert into donor values(0028,'Bania',9898979898,'18@gmailcom','2001-01-
20','xyzabm','AB+'); insert into donor values(0029,'Banno',9898969898,'19@gmailcom','2001-
01-21','xyzaic','B-');

```

The screenshot shows the Oracle SQL Developer interface. The top part is the 'Query 1' editor with the following content:

```

Query 1 X sys_config employee employee employee employee student class class employee SQL File 12*
Find < > [ ] [ ] Limit to 1000 rows [ ] [ ] Jump to SQLAdditions < > [ ] [ ] Done
1 • insert into donor values(0001,'Ram',9898989898,'1@gmailcom','2001-01-01','xyzabc','B+');
2 • insert into donor values(0002,'Ama',9898989898,'2@gmailcom','2001-01-02','xyzabc','A+');
3 • insert into donor values(0003,'jili',9898989898,'3@gmailcom','2001-01-03','xyzabc','o+');
4 • insert into donor values(0004,'Juli',9898989898,'4@gmailcom','2001-01-04','xyzabc','A+');
5 • insert into donor values(0005,'aka',9898989898,'5@gmailcom','2001-01-05','xyzabc','B+');
6 • insert into donor values(0006,'baka',9898989898,'6@gmailcom','2001-01-06','xyzabc','C+');
7 • insert into donor values(0007,'daka',9898989898,'7@gmailcom','2001-01-07','xyzabc','A+');
8 • insert into donor values(0008,'sid',9898989898,'8@gmailcom','2001-01-08','xyzabc','B+');
9 • insert into donor values(0009,'mid',9898989898,'9@gmailcom','2001-01-11','xyzabc','C+');
10 • insert into donor values(0010,'rewa',9898989898,'10@gmailcom','2001-01-12','xyzabc','B+');
11 • insert into donor values(0011,'raywa',9898989898,'11@gmailcom','2001-01-13','xyzabc','AB+');
12 • insert into donor values(0012,'rahwa',9898989898,'12@gmailcom','2001-01-14','xyzabc','O-');
13 • insert into donor values(0013,'gourab',9898989898,'13@gmailcom','2001-01-15','xyzabc','A+');
14 • insert into donor values(0014,'Gaurav',9898989898,'14@gmailcom','2001-01-16','xyzabc','B+');

```

The right side of the interface has a tooltip: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

The bottom part is the 'Output' tab, which displays the results of the executed queries:

Action	Time	Action	Message	Duration / Fetch
53	13:15:41	insert into donor values(0011,'raywa',9898989898,'11@gmailcom','2001-01-13','xyzabc','AB+')	1 row(s) affected	0.000 sec
54	13:15:41	insert into donor values(0012,'rahwa',9898989898,'12@gmailcom','2001-01-14','xyzabc','O-')	1 row(s) affected	0.000 sec
55	13:15:41	insert into donor values(0013,'gourab',9898989898,'13@gmailcom','2001-01-15','xyzabc','A+')	1 row(s) affected	0.000 sec
56	13:15:41	insert into donor values(0014,'Gaurav',9898989898,'14@gmailcom','2001-01-16','xyzabc','B+')	1 row(s) affected	0.000 sec
57	13:15:41	insert into donor values(0015,'Kasisa',9898589898,'15@gmailcom','2001-01-17','xyzabc','AB-')	1 row(s) affected	0.000 sec
58	13:15:41	insert into donor values(0016,'Kosisa',9898589898,'16@gmailcom','2001-01-18','xyzabc','A-')	1 row(s) affected	0.000 sec
59	13:15:41	insert into donor values(0017,'kasola',9898589898,'17@gmailcom','2001-01-19','xyzabc','B+')	1 row(s) affected	0.000 sec
60	13:15:41	insert into donor values(0018,'Banno',9898969898,'18@gmailcom','2001-01-20','xyzabc','AB-')	1 row(s) affected	0.000 sec
61	13:15:41	insert into donor values(0019,'Banno',9898969898,'19@gmailcom','2001-01-21','xyzabc','B-')	1 row(s) affected	0.000 sec

#### RECIPIENT:

```

insert into recipient values(0002,'KAma',9898989898,'2@gmailcom','A+');

insert into recipient values(0001,'KRam',9898189898,'1@gmailcom','b+');

insert into recipient values(0003,'Kjili',9892989898,'3@gmailcom','o+'); insert

into recipient values(0004,'KJuli',9892989898,'4@gmailcom','A+'); insert into

recipient values(0005,'Kaka',9898289898,'5@gmailcom','B+'); insert into

recipient values(0006,'kbaka',9892989898,'6@gmailcom','C+'); insert into

```

```
recipient values(0007,'kdaka',9893989898,'7@gmailcom','A+'); insert into
recipient values(0008,'ksid',9898984898,'8@gmailcom','B+'); insert into
recipient values(0009,'kmid',9898989898,'9@gmailcom','C+'); insert into
recipient values(0010,'krewa',9898939898,'10@gmailcom','B+'); insert into
recipient values(0011,'kraywa',9898389898,'11@gmailcom','AB+'); insert into
recipient values(0012,'krahwa',9898389898,'12@gmailcom','O-'); insert into
recipient values(0013,'kgourab',9893989898,'13@gmailcom','A+'); insert into
recipient values(0014,'kGaurav',98983989898,'14@gmailcom','b+'); insert into
recipient values(0016,'kKosis',9898939898,'16@gmailcom','A-'); insert into
recipient values(0015,'kKasis',9898939898,'15@gmailcom','AB-'); insert into
recipient values(0017,'kkasol',9898989898,'17@gmailcom','B+'); insert into
recipient values(0018,'kBani',9198989898,'18@gmailcom','AB+'); insert into
recipient values(0019,'kBanno',9298989898,'19@gmailcom','B-'); insert into
recipient values(0021,'kRama',9198949898,'1@gmailcom','b+'); insert into
recipient values(0022,'kAmaa',9298949898,'2@gmailcom','A+'); insert into
recipient values(0023,'kjilia',9398289898,'3@gmailcom','o+'); insert into
recipient values(0024,'kJulia',9498589898,'4@gmailcom','A+'); insert into
recipient values(0025,'kakaa',9818969898,'5@gmailcom','B+'); insert into
recipient values(0026,'kbakaa',9198589898,'6@gmailcom','C+'); insert into
recipient values(0027,'kdkakaa',9198889898,'7@gmailcom','A+'); insert into
recipient values(0028,'kskida',9818979898,'8@gmailcom','B+'); insert into
recipient values(0029,'kmida',9891959898,'9@gmailcom','C+'); insert into
recipient values(0020,'krewaa',9818389898,'10@gmailcom','B+'); insert into
recipient values(0021,'kraywaa',9811989898,'11@gmailcom','AB+'); insert into
recipient values(0022,'krahwaa',9891989898,'12@gmailcom','O-'); insert into
recipient values(0023,'kgouraba',9818989898,'13@gmailcom','A+'); insert into
recipient values(0024,'kGaurava',9818989898,'14@gmailcom','b+'); insert into
recipient values(0025,'kKasis',9891589898,'15@gmailcom','AB-'); insert into
recipient values(0026,'kKosisa',9891589898,'16@gmailcom','A-'); insert into
recipient values(0027,'kasola',9198589898,'17@gmailcom','B+'); insert into
recipient values(0028,'kBania',9198979898,'18@gmailcom','AB+');
```

```
insert into recipient values(0029,'kBannoaa',9198969898,'19@gmailcom','B-');
```

```
1 insert into recipient values(0002, 'KAma', 9898989898, '2@gmailcom', 'A+');
2 • insert into recipient values(0001, 'Kram', 9898189898, '1@gmailcom', 'B-');
3 • insert into recipient values(0003, 'Kjili', 9892989898, '3@gmailcom', 'O+');
4 • insert into recipient values(0004, 'KJuli', 9892989898, '4@gmailcom', 'A+');
5 • insert into recipient values(0005, 'Kaka', 9898289898, '5@gmailcom', 'B+');
6 • insert into recipient values(0006, 'kbaka', 9892989898, '6@gmailcom', 'C+');
7 • insert into recipient values(0007, 'kdaka', 9893989898, '7@gmailcom', 'A+');
8 • insert into recipient values(0008, 'ksid', 9898984898, '8@gmailcom', 'B+');
9 • insert into recipient values(0009, 'kmid', 9898989898, '9@gmailcom', 'C+');
10 • insert into recipient values(0010, 'krewa', 9898939898, '10@gmailcom', 'B-');
11 • insert into recipient values(0011, 'kraywa', 9898389898, '11@gmailcom', 'AB-');
12 • insert into recipient values(0012, 'krahwa', 9898389898, '12@gmailcom', 'O-');
13 • insert into recipient values(0013, 'kgourab', 9893989898, '13@gmailcom', 'A+');
14 • insert into recipient values(0014, 'kgaurav', 9898398989, '14@gmailcom', 'B+');
15 • insert into recipient values(0016, 'kKosis', 9898939898, '16@gmailcom', 'A-');
16 • insert into recipient values(0015, 'kkasis', 9898939898, '15@gmailcom', 'AB-');
17 • insert into recipient values(0017, 'kkasol', 9898989898, '17@gmailcom', 'B+');
18 • insert into recipient values(0018, 'kBani', 9198989898, '18@gmailcom', 'AB+');
```

Action	Time	Action	Message	Duration / Fetch
101	13:36:20	insert into recipient values(0028,kukida',9818979898,8@gmailcom','B+)	1 row(s) affected	0.000 sec
102	13:36:20	insert into recipient values(0029,kmida',9891959898,9@gmailcom','C+)	1 row(s) affected	0.000 sec
103	13:36:20	insert into recipient values(0020,krewaa',9818389898,10@gmailcom','B+')	1 row(s) affected	0.016 sec
104	13:36:20	insert into recipient values(0021,kraywa',9811989898,11@gmailcom','AB+')	Error Code: 1062. Duplicate entry '21' for key 'recipient.PRIMARY'	0.000 sec

#### REQUEST:

```
insert into request values(0002,0001); insert
into request values(0001,0002); insert into
request values(0003,0003); insert into
request values(0004,0004); insert into
request values(0005,0005); insert into
request values(0006,0006); insert into
request values(0007,0007); insert into
request values(0008,0008); insert into
request values(0009,0009); insert into
request values(0010,0010); insert into
request values(0011,0011); insert into
request values(0012,0012); insert into
request values(0013,0013); insert into
request values(0014,0014); insert into
request values(0016,0015); insert into
request values(0015,0016); insert into
request values(0017,0017); insert into
request values(0018,0018); insert into
```

```

request values(0019,0019); insert into
request values(0021,0021); insert into
request values(0022,0022); insert into
request values(0023,0023); insert into
request values(0024,0024); insert into
request values(0025,0025); insert into
request values(0026,0026); insert into
request values(0027,0027); insert into
request values(0028,0028);

```

```
insert into request values(0029,0029);
```

The screenshot shows the Oracle SQL Developer interface. The top part is a query editor titled "Query 1" containing the SQL code provided above. The bottom part is an "Output" window titled "Action Output" which displays the results of the executed statements. The results table has columns: #, Time, Action, Message, and Duration / Fetch.

#	Time	Action	Message	Duration / Fetch
134	15:02:35	insert into request values(0026,0026)	1 row(s) affected	0.000 sec
135	15:02:35	insert into request values(0027,0027)	1 row(s) affected	0.000 sec
136	15:02:35	insert into request values(0028,0028)	1 row(s) affected	0.000 sec
137	15:02:35	insert into request values(0029,0029)	1 row(s) affected	0.000 sec

#### ORGANISATION:

```

insert into organisation values(1111,9898989898,'org1@gmail.com','Punjab','TIET','Patiala'); insert
into organisation values(1112,9898989898,'org2@gmail.com','Punjab','Abc road','Ludhiana'); insert
into organisation values(1113,1898989898,'org3@gmail.com','Punjab','BT NAGAR','Amritsar');
insert into organisation values(1114,9298989898,'org4@gmail.com','Punjab','RP ROAD','Bhatinda');

```

```

Query 1 × sys_config employee employee employee student class class employee SQL File 12*
Find Done
1 • insert into organisation values(1111,9898989898,'org1@gmail.com','Punjab','TIET','Patiala');
2 • insert into organisation values(1112,9898989898,'org2@gmail.com','Punjab','Abc road','Ludhiana');
3 • insert into organisation values(1113,1898989898,'org3@gmail.com','Punjab','BT NAGAR','Amritsar');
4 • insert into organisation values(1114,9298989898,'org4@gmail.com','Punjab','RP ROAD','Bhatinda');

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output Context Help Snippets
Action Output
# Time Action Message Duration / Fetch
140 15:12:08 insert into organisation values(1111,9898989898,'org1@gmail.com','Punjab','TIET','Patiala') 1 row(s) affected 0.015 sec
141 15:12:08 insert into organisation values(1112,9898989898,'org2@gmail.com','Punjab','Abc road','Ludhiana') 1 row(s) affected 0.000 sec
142 15:12:08 insert into organisation values(1113,1898989898,'org3@gmail.com','Punjab','BT NAGAR','Amritsar') 1 row(s) affected 0.016 sec
143 15:12:08 insert into organisation values(1114,9298989898,'org4@gmail.com','Punjab','RP ROAD','Bhatinda') 1 row(s) affected 0.000 sec

```

## DONATION:

```

insert into donation values(10,0010,'rewa',9898989898,'10@gmailcom','2001-01-12','xyzabc','B+',1111);

insert into donation values(11,0011,'raywa',9898989898,'11@gmailcom','2001-01-13','xyzabc','AB+',1112);

insert into donation values(12,0012,'rahwa',9898989898,'12@gmailcom','2001-01-14','xyzabc','O-',1113);

insert into donation values(13,0013,'gourab',9898989898,'13@gmailcom','2001-01-15','xyzabc','A+',1114);

insert into donation values(14,0014,'Gaurav',9898989898,'14@gmailcom','2001-01-16','xyzabc','b+',1111);

insert into donation values(15,0015,'Kasis',9898989898,'15@gmailcom','2001-01-17','xyzabc','AB-',1112);

insert into donation values(16,0016,'Kosis',9898989898,'16@gmailcom','2001-01-18','xyzabc','A-',1113);

insert into donation values(17,0017,'kasol',9898989898,'17@gmailcom','2001-0119','xyzabc','B+',1114);

insert into donation values(18,0018,'Bani',9898989898,'18@gmailcom','2001-01-20','xyzabc','AB+',1111);

insert into donation values(19,0019,'Banno',9898989898,'19@gmailcom','2001-01-21','xyzabc','B-',1112);

insert into donation values(20,0021,'Rama',9898949898,'1@gmailcom','2001-01-01','xyzfbc','b+',1113);

insert into donation values(21,0022,'Amaa',9898949898,'2@gmailcom','2001-01-

```

```

02','xyzfbc','A+',1114);

insert into donation values(22,0023,'jilia',9898289898,'3@gmailcom','2001-01-03','xyfabc','o+',1111);

insert into donation values(23,0024,'Julia',9898589898,'4@gmailcom','2001-01-04','xyfabc','A+',1112);

insert into donation values(24,0025,'akaa',9898969898,'5@gmailcom','2001-01-05','xyfabc','B+',1113);

insert into donation values(25,0026,'bakaa',9898589898,'6@gmailcom','2001-01-06','xfzabc','C+',1114);

insert into donation values(26,0027,'dakaa',9898889898,'7@gmailcom','2001-01-07','xyzafc','A+',1111);

insert into donation values(27,0028,'sida',9898979898,'8@gmailcom','2001-01-10','xyzabf','B+',1112);

insert into donation values(28,0029,'mida',9898959898,'9@gmailcom','2001-01-11','xyzafc','C+',1113);

```

The screenshot shows the MySQL Workbench interface with a query editor containing a large block of SQL insert statements. The top pane displays the results of the query, showing 28 rows inserted. The bottom pane shows the action output, which lists 4 rows affected. A tooltip on the right side of the interface provides information about automatic context help.

#	Time	Action	Message	Duration / Fetch
160	15:29:52	insert into donation values(25,0026,'bakaa',9898589898,'6@gmailcom','2001-01-06','xfzabc','C+',1114)	1 row(s) affected	0.000 sec
161	15:29:52	insert into donation values(26,0027,'dakaa',9898889898,'7@gmailcom','2001-01-07','xyzafc','A+',1111)	1 row(s) affected	0.000 sec
162	15:29:52	insert into donation values(27,0028,'sida',9898979898,'8@gmailcom','2001-01-10','xyzabf','B+',1112)	1 row(s) affected	0.000 sec
163	15:29:52	insert into donation values(28,0029,'mida',9898959898,'9@gmailcom','2001-01-11','xyzafc','C+',1113)	1 row(s) affected	0.016 sec

## ORGANIZES:

```

insert into organizes values(1111,21,'2022-01-10','10:00:00','weekly blood donation camp','abc Chowk','camp1');

```

```

insert into organizes values(1112,31,'2022-02-09','11:00:00','monthly blood donation camp','abc Chowk','camp2');

```

```

insert into organizes values(1113,40,'2022-03-08','12:00:00','weekly blood donation camp','ab road','camp3');

```

```

insert into organizes values(1114,50,'2022-04-07','13:00:00','quarterly blood donation camp','aadarsh nagar','camp4');

insert into organizes values(1111,22,'2022-05-06','14:00:00','weekly blood donation camp','rc colony','camp5');

insert into organizes values(1112,32,'2022-06-05','14:00:00','monthly blood donation camp','op road','camp6');

insert into organizes values(1113,41,'2022-07-04','11:00:00','weekly blood donation camp','nabha road','camp7');

insert into organizes values(1114,51,'2022-08-03','12:00:00','yearly blood donation camp','tiet road','camp8');

insert into organizes values(1111,23,'2022-09-02','13:00:00','weekly blood donation camp','a colony','camp9');

insert into organizes values(1112,33,'2022-10-01','14:00:00','monthly blood donation camp','akas homes','camp10');

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or toggle automatic help.

#	Time	Action	Message	Duration / Fetch
174	15:44:32	insert into organizes values(1113,41,2022-07-04,'11:00:00','weekly blood donation camp','nabha road','camp7')	1 row(s) affected	0.000 sec
175	15:44:32	insert into organizes values(1114,51,2022-08-03,'12:00:00','yearly blood donation camp','tiet road','camp8')	1 row(s) affected	0.000 sec
176	15:44:32	insert into organizes values(1111,23,2022-09-02,'13:00:00','weekly blood donation camp','a colony','camp9')	1 row(s) affected	0.000 sec
177	15:44:32	insert into organizes values(1112,33,2022-10-01,'14:00:00','monthly blood donation camp','akas homes','camp10')	1 row(s) affected	0.000 sec

## INVENTORY:

```

insert into inventory values(1111,100,'B+','2021-05-01',2);

insert into inventory values(1112,101,'AB+','2021-05-04',1);

insert into inventory values(1113,102,'AB-','2021-05-06',1);

insert into inventory values(1114,103,'B+','2021-03-01',1);

insert into inventory values(1111,104,'AB-','2021-05-10',1);

insert into inventory values(1112,105,'B+','2021-03-01',2);

insert into inventory values(1113,106,'B+','2021-05-20',2);

insert into inventory values(1114,107,'B-','2021-05-01',2); insert

```

```
into inventory values(1111,108,'AB+','2021-02-01',1); insert into
inventory values(1112,109,'o-','2021-05-30',1); insert into
inventory values(1113,110,'b+','2021-02-03',2); insert into
inventory values(1114,111,'AB-','2021-04-25',1); insert into
inventory values(1111,112,'b+','2021-05-11',2); insert into
inventory values(1112,113,'o+','2021-05-01',1); insert into
inventory values(1113,114,'b-','2021-05-12',1); insert into
inventory values(1114,115,'o+','2021-06-13',2); insert into
inventory values(1111,116,'AB+','2021-05-01',1); insert into
inventory values(1112,117,'o+','2021-03-01',2); insert into
inventory values(1113,118,'b-','2021-05-16',1); insert into
inventory values(1114,119,'AB+','2021-05-01',1); insert into
inventory values(1111,120,'o-','2021-02-20',2); insert into
inventory values(1112,121,'b+','2021-05-01',1); insert into
inventory values(1113,122,'b-','2021-02-11',1); insert into
inventory values(1114,123,'AB+','2021-05-01',1); insert into
inventory values(1111,124,'o+','2021-05-13',1); insert into
inventory values(1112,125,'b+','2021-03-22',2); insert into
inventory values(1113,126,'o-','2021-05-23',1); insert into
inventory values(1114,127,'AB+','2021-04-08',1); insert into
inventory values(1111,128,'o-','2021-05-09',1);
```

```

Query 1 × sys_config employee employee employee employee student class class employee SQL File 12*
Find Done
1 • insert into inventory values(1111,100,'B+', '2021-05-01',2);
2 • insert into inventory values(1112,101,'AB+', '2021-05-04',1);
3 • insert into inventory values(1113,102,'AB-', '2021-05-06',1);
4 • insert into inventory values(1114,103,'b+', '2021-03-01',1);
5 • insert into inventory values(1111,104,'AB-', '2021-05-10',1);
6 • insert into inventory values(1112,105,'b+', '2021-03-01',2);
7 • insert into inventory values(1113,106,'b+', '2021-05-20',2);
8 • insert into inventory values(1114,107,'b-', '2021-05-01',2);
9 • insert into inventory values(1111,108,'AB+', '2021-02-01',1);
10 • insert into inventory values(1112,109,'o-', '2021-05-30',1);
11 • insert into inventory values(1113,110,'b+', '2021-02-03',2);
12 • insert into inventory values(1114,111,'AB-', '2021-04-25',1);
13 • insert into inventory values(1111,112,'b+', '2021-05-11',2);
14 • insert into inventory values(1112,113,'o+', '2021-05-01',1);
15 • insert into inventory values(1113,114,'b+-', '2021-05-12',1);
16 • insert into inventory values(1114,115,'o+', '2021-06-13',2);
17 • insert into inventory values(1111,116,'AB+', '2021-05-01',1);
18 • insert into inventory values(1112,117,'o+', '2021-03-01',2);

Output
Action Output
* Time Action Message Duration / Fetch
204 16:04:32 insert into inventory values(1112,125,b+,'2021-03-22',2) 1 row(s) affected 0.000 sec
205 16:04:32 insert into inventory values(1113,126,o-,'2021-05-23',1) 1 row(s) affected 0.000 sec
206 16:04:32 insert into inventory values(1114,127,AB+,'2021-04-08',1) 1 row(s) affected 0.000 sec
207 16:04:32 insert into inventory values(1111,128,o-,'2021-05-09',1) 1 row(s) affected 0.000 sec

```

BLOOD:

```

insert into blood values(100,'B+', '2021-05-01',2);
insert into blood values(101,'AB+', '2021-05-04',1);
insert into blood values(102,'AB-', '2021-05-06',1);
insert into blood values(103,'b+', '2021-03-01',1);
insert into blood values(104,'AB-', '2021-05-10',1);
insert into blood values(105,'b+', '2021-03-01',2);
insert into blood values(106,'b+', '2021-05-20',2);
insert into blood values(107,'b-', '2021-05-01',2); insert
into blood values(108,'AB+', '2021-02-01',1); insert into
blood values(109,'o-', '2021-05-30',1); insert into blood
values(110,'b+', '2021-02-03',2); insert into blood
values(111,'AB-', '2021-04-25',1); insert into blood
values(112,'b+', '2021-05-11',2); insert into blood
values(113,'o+', '2021-05-01',1); insert into blood
values(114,'b+-', '2021-05-12',1); insert into blood
values(115,'o+', '2021-06-13',2); insert into blood
values(116,'AB+', '2021-05-01',1); insert into blood
values(117,'o+', '2021-03-01',2); insert into blood
values(118,'b-', '2021-05-16',1); insert into blood

```

```

values(119,'AB+','2021-05-01',1); insert into blood
values(120,'o-','2021-02-20',2); insert into blood
values(121,'b+','2021-05-01',1); insert into blood
values(122,'b-','2021-02-11',1); insert into blood
values(123,'AB+','2021-05-01',1); insert into blood
values(124,'o+','2021-05-13',1);

```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains 18 numbered SQL statements for inserting data into the 'blood' table.
- Output Pane:** Shows the results of the executed queries, with each row indicating 1 row(s) affected.
- Help Tooltip:** A tooltip on the right side of the interface provides instructions on how to enable automatic context help.

Action	Time	Message	Duration / Fetch
235	16:12:13	insert into blood values(121,b+,'2021-05-01',1)	0.016 sec
236	16:12:13	insert into blood values(122,b-,'2021-02-11',1)	0.000 sec
237	16:12:13	insert into blood values(123,AB+,'2021-05-01',1)	0.000 sec
238	16:12:13	insert into blood values(124,o+,'2021-05-13',1)	0.000 sec

## RECIPIENT1

```

insert into recipient1 values('A+',1001,'1@gmail.com','ram',6565656565,100); insert
into recipient1 values('O+',1002,'2@gmail.com','Shayam',6565656565,101); insert
into recipient1 values('AB+',1003,'3@gmail.com','Shyama',6565656565,102); insert
into recipient1 values('A+',1004,'4@gmail.com','rama',6565656565,103); insert into
recipient1 values('B+',1005,'5@gmail.com','Harsh',6565656565,104); insert into
recipient1 values('O+',1006,'6@gmail.com','Harsha',6565636565,105); insert into
recipient1 values('AO+',1007,'7@gmail.com','Hari',6565616565,106); insert into
recipient1 values('AB+',1008,'8@gmail.com','Haria',6562656565,107); insert into
recipient1 values('A+',1009,'9@gmail.com','Rahul',6565656565,108); insert into
recipient1 values('AB+',1010,'10@gmail.com','Rohit',6565656565,109); insert into
recipient1 values('B+',1011,'11@gmail.com','Raghav',6565656165,110); insert into
recipient1 values('O+',1012,'12@gmail.com','Ram',6565656561,111); insert into
recipient1 values('AB+',1013,'13@gmail.com','Rohit',6565651565,112); insert into

```

```

recipient1 values('O+',1014,'14@gmail.com','Raman',6565651565,113); insert into
recipient1 values('AO+',1015,'15@gmail.com','Gaurav',6561656565,114); insert into
recipient1 values('AO+',1016,'16@gmail.com','Anamik',6562656565,115); insert into
recipient1 values('BA+',1017,'17@gmail.com','Ana',6565656265,116); insert into
recipient1 values('B+',1018,'18@gmail.com','Aman',6565656365,117); insert into
recipient1 values('B+',1020,'19@gmail.com','Amanik',6565456565,118); insert into
recipient1 values('B+',1021,'20@gmail.com','Ram',6565456565,119); insert into
recipient1 values('B+',1022,'21@gmail.com','ram',6515656565,120); insert into
recipient1 values('B+',1023,'22@gmail.com','ram',6265656565,121); insert into
recipient1 values('B+',1024,'23@gmail.com','ram',6565636565,122);

insert into recipient1 values('B+',1025,'24@gmail.com','ram',6565636565,123);

```

The screenshot shows the Oracle SQL Developer interface. The top part is a query editor window titled "Query 1" with tabs for "sys\_config" and various schema objects like "employee", "student", "class". The main area contains the SQL code for inserting data into the "recipient1" table. The bottom part is an "Output" window titled "Action Output" which displays the results of the executed queries, showing 3 rows affected for each of the 25 insert statements.

```

Query 1 × sys_config employee employee employee student class class employee SQL File 12*
Find ▾ [ ] Done
1 • insert into recipient1 values('A+',1001,'1@gmail.com','ram',6565656565,100);
2 • insert into recipient1 values('O+',1002,'2@gmail.com','Shyam',6565656565,101);
3 • insert into recipient1 values('AB+',1003,'3@gmail.com','Shyama',6565656565,102);
4 • insert into recipient1 values('A+',1004,'4@gmail.com','rama',6565656565,103);
5 • insert into recipient1 values('B+',1005,'5@gmail.com','Harsh',6565656565,104);
6 • insert into recipient1 values('O+',1006,'6@gmail.com','Harsha',6565636565,105);
7 • insert into recipient1 values('AO+',1007,'7@gmail.com','Hari',6565616565,106);
8 • insert into recipient1 values('AB+',1008,'8@gmail.com','Maria',6562656565,107);
9 • insert into recipient1 values('A+',1009,'9@gmail.com','Rahul',6565656565,108);
10 • insert into recipient1 values('AB+',1010,'10@gmail.com','Rohit',6565656565,109);
11 • insert into recipient1 values('B+',1011,'11@gmail.com','Raghav',6565656165,110);
12 • insert into recipient1 values('O+',1012,'12@gmail.com','Ram',6565656561,111);
13 • insert into recipient1 values('AB+',1013,'13@gmail.com','Rohit',6565651565,112);
14 • insert into recipient1 values('O+',1014,'14@gmail.com','Raman',6565651565,113);
15 • insert into recipient1 values('AO+',1015,'15@gmail.com','Gaurav',6561656565,114);
16 • insert into recipient1 values('AO+',1016,'16@gmail.com','Anamik',6562656565,115);
17 • insert into recipient1 values('BA+',1017,'17@gmail.com','Ana',6565656265,116);
18 • insert into recipient1 values('B+',1018,'18@gmail.com','Aman',6565636565,117);

Context Help Snippets
Output
Action Output
# Time Action Message Duration / I
248 16:29:04 insert into recipient1 values('AO+',1007,'7@gmail.com','Hari',6565616565,106) 1 row(s) affected 0.000 sec
249 16:29:04 insert into recipient1 values('AB+',1008,'8@gmail.com','Hana',6562656565,107) 1 row(s) affected 0.000 sec
250 16:29:04 insert into recipient1 values('A+',1009,'9@gmail.com','Rahul',6565656565,108) 1 row(s) affected 0.000 sec

```

# SQL/PLSQL

## QUERIES

1.FIND THE NAME OF DONOR FOR RECIPIENT WHO RECEIVED BLOOD FROM UID 10!

select name from donor where uid in ( select uid1 from request where uid2 in ( select uid from recipient where uid=10))

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Query 1" and contains the following SQL code:

```
1 • select name from donor where uid in ( select uid1 from request where uid2 in ( select uid from recipient where uid=10))
2
```

The bottom window is titled "Result Grid" and displays the results of the query:

name
rewa

2.FIND THE NAME AND PHONE NUMBER OF DONOR FOR RECIPIENT WHO RECEIVED BLOOD FROM UID 5!

select name,phone\_no from donor where uid in ( select uid1 from request where uid2 in ( select uid from recipient where uid=5))

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Query 1" and contains the following SQL code:

```
1 • select name,phone_no from donor where uid in ( select uid1 from request where uid2 in ( select uid from recipient where uid=5))
2
```

The bottom window is titled "Result Grid" and displays the results of the query:

name	phone_no
aka	9898989898

### 3.DISPLAY THE NAMES AND CONTACT OF THE PEOPLE WHO HAVE DONATED BLOOD IN PATIALA CITY.

SELECT name, phone\_no from donation where oid1 in (select oid from organisation where city="Patiala");

The screenshot shows a database interface with a query editor and a result grid. The query is:

```
1 •  SELECT name, phone_no from donation where oid1 in (select oid from organisation where city="Patiala")
```

The result grid displays the following data:

name	phone_no
rewa	9898989898
Gaurav	9898989898
Bani	9898989898
jilia	9898289898
dakaa	9898889898

Contextual help on the right side of the interface states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

### 4.COUNT THE NUMBER OF DONATIONS IN AMRITSAR CITY

select count(\*) from donation where oid1 in (select oid from organisation where city="Amritsar")

The screenshot shows a database interface with a query editor and a result grid. The query is:

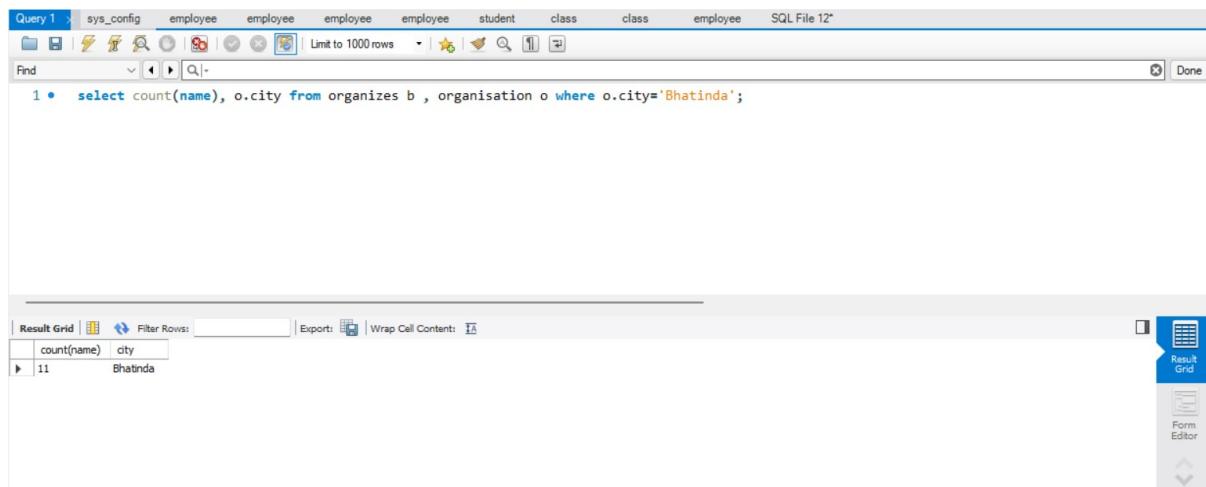
```
1 •  select count(*) from donation where oid1 in (select oid from organisation where city="Amritsar")
```

The result grid displays the following data:

count(*)
5

### 5.FIND TOTAL NUMBER OF CAMPS IN BHATINDA CITY

```
select count(name), o.city from organizes b , organisation o where o.city='Bhatinda';
```



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

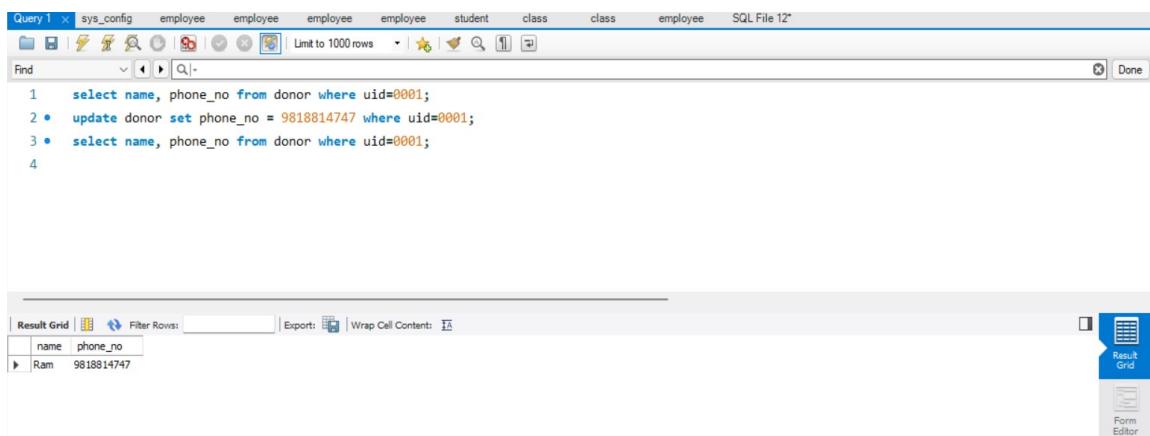
```
1 •  select count(name), o.city from organizes b , organisation o where o.city='Bhatinda';
```

The results are displayed in a "Result Grid" table:

count(name)	city
11	Bhatinda

## 6.UPDATE PHONE NUMBER OF DONOR WITH ID 0001:

```
select name, phone_no from donor where uid=0001; update  
donor set phone_no = 9818814747 where uid=0001; select  
name, phone_no from donor where uid=0001;
```



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The queries are:

```
1 •  select name, phone_no from donor where uid=0001;  
2 •  update donor set phone_no = 9818814747 where uid=0001;  
3 •  select name, phone_no from donor where uid=0001;  
4
```

The results are displayed in a "Result Grid" table:

name	phone_no
Ram	9818814747

## 7.DISPLAY TOTAL NUMBER OF DONATIONS BLOOD GROUP WISE

```
select blood_group,count(certificate_id) from donation group by blood_group;
```

The screenshot shows a SQL query window with the following content:

```
1 select blood_group, count(certificate_id) from donation group by blood_group;
```

The results grid displays the following data:

blood_group	count(certificate_id)
B+	6
AB+	2
O-	1
A+	4
A-	1
B-	1
O+	1

The output pane shows the following log entries:

#	Time	Action	Message
1	10:50:35	use blood_bank	0 row(s) affected
2	10:54:54	desc donation	9 row(s) returned
3	10:55:24	select count(certificate_id) from donation LIMIT 0, 1000	1 row(s) returned
4	10:56:11	desc donation	9 row(s) returned
5	10:56:43	select blood_group, count(certificate_id) from donation group by blood_group LIMIT 0, 1000	9 row(s) returned

8.DISPLAY TOTAL AMOUNT OF B+ BLOOD COLLECTRRD FROM PATIALA CITY BY ANY ORGANISATION/CAMP.

The screenshot shows a SQL query window with the following content:

```
1 select sum(amount) from inventory where btype='B+' and oid1=1111;
```

The results grid displays the following data:

sum(amount)
4

The output pane shows the following log entries:

#	Time	Action	Message
9	11:05:44	select sum(amount) from inventory where btype='B+' and oid1='1111' LIMIT 0, 1000	Error Code: 1054. Unknown column 'oid1' in 'where clause'
10	11:06:06	select sum(amount) from inventory where btype='B+' and oid=1111 LIMIT 0, 1000	Error Code: 1054. Unknown column 'oid' in 'where clause'
11	11:06:18	desc inventory	5 row(s) returned
12	11:06:25	1	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1' at line 1
13	11:06:36	select sum(amount) from inventory where btype='B+' and oid1=1111 LIMIT 0, 1000	1 row(s) returned

9.DISPLAY ALL BLOOD SAMPLES COLLECTED IN 5<sup>TH</sup> MONTH IN THE INVENTORY

SELECT \* FROM inventory where date1>"2021-05-00" and date1<"2021-05-31";

```

1  SELECT * FROM inventory where date1>"2021-05-00" and date1<"2021-05-31";

```

Result Grid

oid1	bloodid	btype	date1	amount
1111	100	B+	2021-05-01	2
1112	101	AB+	2021-05-04	1
1113	102	AB-	2021-05-06	1
1114	104	AB-	2021-05-08	1
1113	106	b+	2021-05-20	1
1114	107	b-	2021-05-01	2
1112	109	o-	2021-05-30	2
1111	112	b+	2021-05-11	2

Action Output

#	Time	Action	Message
20	11:11:57	SELECT date1 FROM inventory where date1>"2021-03-03" LIMIT 0, 1000	0 row(s) returned
21	11:12:13	SELECT date1 FROM inventory where date1>"2021-05-11" LIMIT 0, 1000	1 row(s) returned
22	11:12:23	SELECT `*` FROM inventory where date1>"2021-05-11" LIMIT 0, 1000	1 row(s) returned
23	11:12:34	SELECT `*` FROM inventory where date1>"2021-05-11" LIMIT 0, 1000	7 row(s) returned
24	11:13:21	SELECT `*` FROM inventory where date1>"2021-05-00" and date1<"2021-05-31" LIMIT 0, 1000	18 row(s) returned

## 10.DISPLAY COUNT OF TOTAL SAMPLES CITY WISE.

```
select oid1, count(amount) from inventory group by oid1;
```

```

1  select oid1, count(amount) from inventory group by oid1;

```

Result Grid

oid1	count(amount)
1111	8
1112	7
1113	7
1114	7

Action Output

#	Time	Action	Message
28	11:18:28	select oid1, count(amount) from inventory group by oid1 LIMIT 0, 1000	4 row(s) returned
29	11:19:36	select o.city, count(amount) from inventory, organisation o group by oid1 LIMIT 0, 1000	4 row(s) returned
30	11:20:33	select oid, count(amount) from inventory group by city LIMIT 0, 1000	Error Code: 1054. Unknown column 'oid' in field list'
31	11:20:34	select oid, count(amount) from inventory group by city LIMIT 0, 1000	Error Code: 1054. Unknown column 'oid' in field list'
32	11:20:52	select oid1, count(amount) from inventory group by oid1 LIMIT 0, 1000	4 row(s) returned

# PLSQL

1. Write a code to check whether a given blood sample is in stock and if yes display to user else raise an exception.

```
declare
ab blood.btype%type; begin
select btype into ab from blood where bloodid=110;
dbms_output.put_line('AMPLE FOUND IN STOCK AND THE REQUESTED
BLOOD SAMPLE IS'||ab);
exception
when too_many_rows then
dbms_output.put_line('query returned more than one record'); end;
```

SQL Worksheet

Clear Find Actions ▾ Save

```
101 -- bloodgroup varchar2(5),
102 -- usid integer primary key,
103 -- email varchar2(30),
104 -- name varchar2(20),
105 -- phone number(10),
106 -- bid1 integer,
107 -- foreign key(bid1) references blood(bloodid));
108
109 -- insert into recipient1 values('A+',1001,'1@gmail.com','ram',6565656565,100);
110
111 declare
112 ab blood.btype%type;
113
114 begin
115 select btype into ab from blood where bloodid=110;
116 dbms_output.put_line('AMPLE FOUND IN STOCK AND THE REQUESTED BLOOD SAMPLE IS'||ab);
117 exception
118 when too_many_rows then
119 dbms_output.put_line('query returned more than one record');
120 end;
121
```

Statement processed.  
AMPLE FOUND IN STOCK AND THE REQUESTED BLOOD SAMPLE IS b+

2. WRITE A CODE TO CHECK IF THE USER HAS ENTERED VALUES ACCORDING TO THE CORRECT DATA TYPE

```
begin
insert into donor
values(0090,'mridul','9118959898A','mridul@gmailcom','200101-
11','xyzafc','C+');

exception
when invalid_number then
dbms_output.put_line('Conversion of string to number failed ');
end;
```

SQL Worksheet

```

112 -- ab_blood.btype%type;
113
114 -- begin
115 -- select btype into ab from blood where bloodid=110;
116 -- dbms_output.put_line('AMPLE FOUND IN STOCK AND THE REQUESTED BLOOD SAMPLE IS '||ab);
117 -- exception
118 -- when too_many_rows then
119 -- dbms_output.put_line('query returned more than one record');
120 -- end;
121
122 begin
123 insert into donor values(0090,'mridul','9118959898A','mridul@gmailcom','2001-01-11','xyzafc','C+');
124
125 |
126 exception
127 when invalid_number then
128 dbms_output.put_line('Conversion of string to number failed ');
129 end;
130
131
132

```

Statement processed.  
Conversion of string to number failed

## USING CURSOR:

### 3.UPDATE THE DONOR DATA BUT FIRST CHECK IF THE DONOR EXISTES ELSE RAISE EXCEPTION

```

declare
total_rows number(2);

begin

update donor
set phone_no = 987456321124 where usid = 0014;
if sql%notfound then dbms_output.put_line(' NO SUCH DONOR FOUND ');
elsif sql%found then total_rows:=sql%rowcount;
dbms_output.put_line('rows affected
'||total_rows); end if; end;

```

SQL Worksheet

```

124 -- insert into donor values(0090,'mridul','9118959898A','mridul@gmailcom','2001-01-11','xyzafc','C+');
125
126 -- exception
127 -- when invalid_number then
128 dbms_output.put_line('Conversion of string to number failed ');
129 -- end;
130
131 declare
132 total_rows number(2);
133
134 begin
135
136 update donor
137 set phone_no = 987456321124 where usid = 0014;
138 if sql%notfound then dbms_output.put_line(' NO SUCH DONOR FOUND ');
139 elsif sql%found then total_rows:=sql%rowcount;
140 dbms_output.put_line('rows affected'||total_rows);
141 end if;
142 end;
143
144

```

Statement processed.  
rows affected 1

### 4.Write a procedure to return the people with a particular blood type and names starting from 'k' and 'r'

```
create or replace procedure blood_filter(b in varchar,n in varchar) is
```

```

bg donor.blood_group%type;
np donor.name%type; id
donor.usid%type; cursor cf is
select usid, name, blood_group from donor where blood_group=b and n LIKE 'r%' or n
LIKE 'R%' or n LIKE 'k%' or n LIKE 'K%'; begin open cf; loop fetch cf into
id,np,bg; exit when cf%notfound; dbms_output.put_line(np|| ' with id = ' ||
id || ' has blood group ' || bg); end loop; close cf; exception
when no_data_found then
dbms_output.put_line('Sorry no such products exist'); end;

```

SQL Worksheet

```

1 create or replace procedure blood_filter(b in varchar,n in varchar)
2 is
3 bg donor.blood_group%type;
4 np donor.name%type;
5 id donor.usid%type;
6 cursor cf is
7 select usid, name, blood_group from donor where blood_group=b and n LIKE 'r%' or n
LIKE 'R%' or n LIKE 'k%' or n LIKE 'K%'; begin open cf; loop fetch cf into
8 id,np,bg;
9 exit when cf%notfound;
10 dbms_output.put_line(np|| ' with id = ' || id || ' has blood group ' || bg);
11 end loop;
12 close cf;
13 exception
14 when no_data_found then
15 dbms_output.put_line('Sorry no such products exist');
16 end;
17
18
19
20

```

Procedure created.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0 · Database Documentation · Ask Tom · Dev Gym  
Built with ❤ using Oracle APEX · Privacy · Terms of Use

5. Write a procedure to check if there is an organisation in the given city else raise an exception

```

create or replace procedure org_details(c in varchar) is
org_city varchar2(20); begin
select city into org_city from organisation where city = c; exception
when no_data_found then
dbms_output.put_line('Sorry no such city exists !!'); end;

```

SQL Worksheet

```

1 create or replace procedure blood_filter(b in varchar,n in varchar)
2 is
3   bg donor.blood_group%type;
4   np donor.name%type;
5   id donor.usid%type;
6   cursor cf is
7   select usid, name, blood_group from donor where blood_group=b and n LIKE 'r%' or n LIKE 'R%' or n LIKE 'k%' or n LIKE 'K%';
8   begin
9     open cf;
10    loop
11      fetch cf into id,np,bg;
12      exit when cf%notfound;
13      dbms_output.put_line(np|| ' with id - ' || id || ' has blood group ' || bg);
14    end loop;
15    close cf;
16  exception
17  when no_data_found then
18    dbms_output.put_line('Sorry no such products exist');
19  end;
20

```

Procedure created.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

## 6. Write a procedure to check if there exists a certificate ID and its not duplicate

create or replace procedure certification(i in integer) is  
 c\_id integer; begin

```

select certificate_id into c_id from donation where certificate_id=i; exception
when no_data_found then
dbms_output.put_line('Sorry no such certificate exists !!');
when too_many_rows then
dbms_output.put_line('Duplicate certificates exist'); end;

```

Home

SQL Worksheet

```

1 create or replace procedure certification(i in integer)
2 is
3   c_id integer;
4   begin
5   select certificate_id into c_id from donation where certificate_id=i;
6   exception
7   when no_data_found then
8     dbms_output.put_line('Sorry no such certificate exists !!');
9   when too_many_rows then
10    dbms_output.put_line('Duplicate certificates exist');
11  end;
12

```

Procedure created.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

## 7. Trigger to count the number of distinct cities before inserting, deleting or updating anything in the organisation table

```

create or replace trigger org_donor before
insert or delete or update on organisation
declare count1 number(3); begin
select count( distinct state) into count1 from organisation;

```

```
dbms_output.put_line(count1);
end;
```

```

1 create table organisation(
2   oid int primary key,
3   contact_id number(10),
4   state varchar2(20),
5   address varchar2(50),
6   city varchar2(20));
7
8   insert into organisation values(111,9898989898,'org1@gmail.com','Punjab','TIEET','Patiala');
9   insert into organisation values(112,9898989898,'org1@gmail.com','Haryana','TIEET','Patiala');
10  insert into organisation values(113,9898989898,'org1@gmail.com','Himachal','TIEET','Patiala');
11  insert into organisation values(114,9898989898,'org1@gmail.com','Punjab','TIEET','Patiala');
12  insert into organisation values(115,9898989898,'org1@gmail.com','Punjab','TIEET','Patiala');
13
14
15 create or replace trigger org_donor
16 before insert or delete or update on organisation
17 declare
18   count number(3);
19 begin
20   select count(distinct state) into count1 from organisation;
21   dbms_output.put_line(count1);
22 end;
23
24
1 row(s) inserted.
3

```

## 8. Trigger to count the number of donors before adding new donors

create or replace trigger count\_donor before

insert or delete or update on donor for

each row declare count1 number(3); begin

select count(\*) into count1 from donor;

dbms\_output.put\_line(count1); end;

SQL Worksheet

```

1 create table donor(
2     usid int PRIMARY KEY,
3     name varchar2(20),
4     phone_no number(10),
5     email varchar2(20),
6     DOB date,
7     pwd varchar(30),
8     blood_group varchar(5));
9
10 create table recipient(
11    usid int primary key,
12    name varchar2(20),
13    phone_no number(10),
14    email varchar(20),
15    blood_group varchar2(5));
16
17
18 insert into donor values(0001,'Ram',9898989898,'1@gmailcom',to_date('2001-01-01','yyyy-mm-dd'),'xyzabc','b+');
19 insert into donor values(0002,'Ram',9898989898,'1@gmailcom',to_date('2001-02-01','yyyy-mm-dd'),'xyzabc','O+');
20 insert into donor values(0003,'Ram',9898989898,'1@gmailcom',to_date('2001-03-01','yyyy-mm-dd'),'xyzabc','A+');
21 insert into donor values(0004,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
22 insert into donor values(0005,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
23 insert into donor values(0006,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
24 insert into donor values(0007,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
25
26 insert into recipient values(0002,'kBania',9198979898,'18@gmailcom','AB+');
27 insert into recipient values(0029,'kBanno',9198969898,'19@gmailcom','B-');
28
29 create or replace trigger count_donor
30 before insert or delete or update on donor
31 for each row
32 declare
33 count1 number(3);
34 begin
35 select count(*) into count1 from donor;
36 dbms_output.put_line(count1);
37 end;
38
39 alter trigger count_donor disable;

```

1 row(s) inserted.  
5

Activate Windows  
Go to Settings to activate Windows.

© 2022 Oracle - Live SQL 22.1.3. running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom · Dev Gym  
Built with ❤ using Oracle APEX · Privacy · Terms of Use

SQL Worksheet

```

19 insert into donor values(0002,'Ram',9898989898,'1@gmailcom',to_date('2001-02-01','yyyy-mm-dd'),'xyzabc','O+');
20 insert into donor values(0003,'Ram',9898989898,'1@gmailcom',to_date('2001-03-01','yyyy-mm-dd'),'xyzabc','A+');
21 insert into donor values(0004,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
22 insert into donor values(0005,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
23 insert into donor values(0006,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
24 insert into donor values(0007,'Ram',9898989898,'1@gmailcom',to_date('2001-04-01','yyyy-mm-dd'),'xyzabc','O-');
25
26 insert into recipient values(0002,'kBania',9198979898,'18@gmailcom','AB+');
27 insert into recipient values(0029,'kBanno',9198969898,'19@gmailcom','B-');
28
29 create or replace trigger count_donor
30 before insert or delete or update on donor
31 for each row
32 declare
33 count1 number(3);
34 begin
35 select count(*) into count1 from donor;
36 dbms_output.put_line(count1);
37 end;
38
39 alter trigger count_donor disable;

```

1 row(s) inserted.  
5

Activate Windows  
Go to Settings to activate Windows.

© 2022 Oracle - Live SQL 22.1.3. running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom · Dev Gym  
Built with ❤ using Oracle APEX · Privacy · Terms of Use