



**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

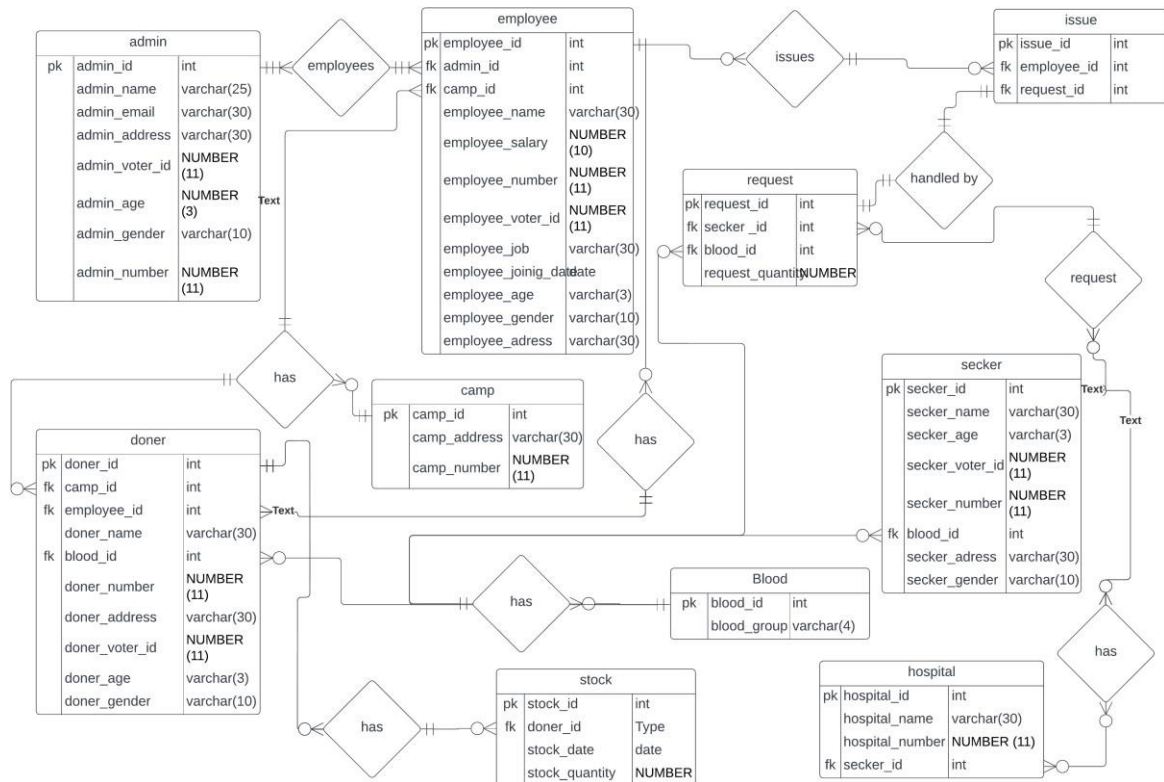
**INTRODUCTION TO DATABASE**

**SPRING 21-22**

1. MD. ALAMIN HOSSEN	21-44943-2	
----------------------	------------	--

# **Blood Bank Management** **System**

# Er diagram



## Description:

Employees are employed by admin. Every employee is allocated to one of the camps. Every camp is overseen by a manager, as well as a Accountant, an Cashier, and nurses. Every camp has its own unique identifier. A blood donor can donate blood in the camp of his choice. If a donor want to donate blood, he must go to a camp and do so through a nurses or employee. A secker may make a request. Secker may request through a hospital or not. Each Every blood type has its own identification number, and all blood is kept in stock. Each stock has its own unique identifier and contains each quantity of blood Date of the stock. Employees can check the inventory and authorize issue requests for blood.

## Total Tables for the Project:

[ Table title: admin ]    admin\_id, admin\_name, admin\_email,  
admin\_address,            admin\_voter\_id,            admin\_age,  
admin\_gender, admin\_number

[ Table title: camp]            camp\_id,  
camp\_address, camp\_number

[ Table title: employee]    employee\_id, admin\_id, camp\_id,  
employee\_name, employee\_salary, employee\_number,  
employee\_voter\_id,            employee\_job,  
employee\_joinig\_date, employee\_age, employee\_gender,  
employee\_adress

[ Table title: Blood]  
blood\_id, blood\_group

[ Table title: stock]            stock\_id, doner\_id,  
stock\_date, stock\_quantity [ Table title: doner]  
doner\_id, camp\_id, employee\_id, doner\_name,  
blood\_id,    doner\_number,    doner\_address,  
doner\_voter\_id, doner\_age doner\_gender

[ Table title: secker]     secker\_id, secker\_name,  
secker\_age, secker\_voter\_id, secker\_number, blood\_id,  
secker\_adress, secker\_gender

[ Table title: hospital]     hospital\_id, hospital\_name,  
hospital\_number, secker\_id

[ Table title: request]     request\_id, secker\_id,  
blood\_id, request\_quantity

[ Table title: issue]     issue\_id,  
employee\_id, request\_id

\*Red front\_color and underline => Primary\_key

\*Violet front\_color => Foreign\_key

## Normalization

1nf : No Multi Valued Attributes.

2nf : No Partial Functional dependency.

3nf : No transitive dependencies found.

## Table Creationwith constraints:

### 1. Admin table:

```
CREATE TABLE admin ( admin_id
int not null, admin_name
varchar(25), admin_email
varchar(30), admin_address
varchar(30), admin_voter_id
NUMBER (11), admin_age
NUMBER (3), admin_gender
varchar(5), admin_number
NUMBER (11),
constraint admin_id primary key (admin_id)
);
```

```
CREATE SEQUENCE admin_id
increment by 1
start with 1
maxvalue 100
nocache
nocycle;
```

0.062 seconds

Worksheet Query Builder

```

CREATE TABLE admin (
  admin_id int not null,
  admin_name varchar(25),
  admin_email varchar(30),
  admin_address varchar(30),
  admin_voter_id NUMBER (11),
  admin_age NUMBER (3),
  admin_gender varchar(10),
  admin_number NUMBER (11),
  constraint admin_id primary key (admin_id)
);

DESC admin;

CREATE SEQUENCE admin_id
  increment by 1

```

Script Output x

Task completed in 0.062 seconds

Name	Null?	Type
ADMIN_ID	NOT NULL	NUMBER(38)
ADMIN_NAME		VARCHAR2(25)
ADMIN_EMAIL		VARCHAR2(30)
ADMIN_ADDRESS		VARCHAR2(30)
ADMIN_VOTER_ID		NUMBER(11)
ADMIN_AGE		NUMBER(3)
ADMIN_GENDER		VARCHAR2(10)
ADMIN_NUMBER		NUMBER(11)

## 2. Camp table:

```

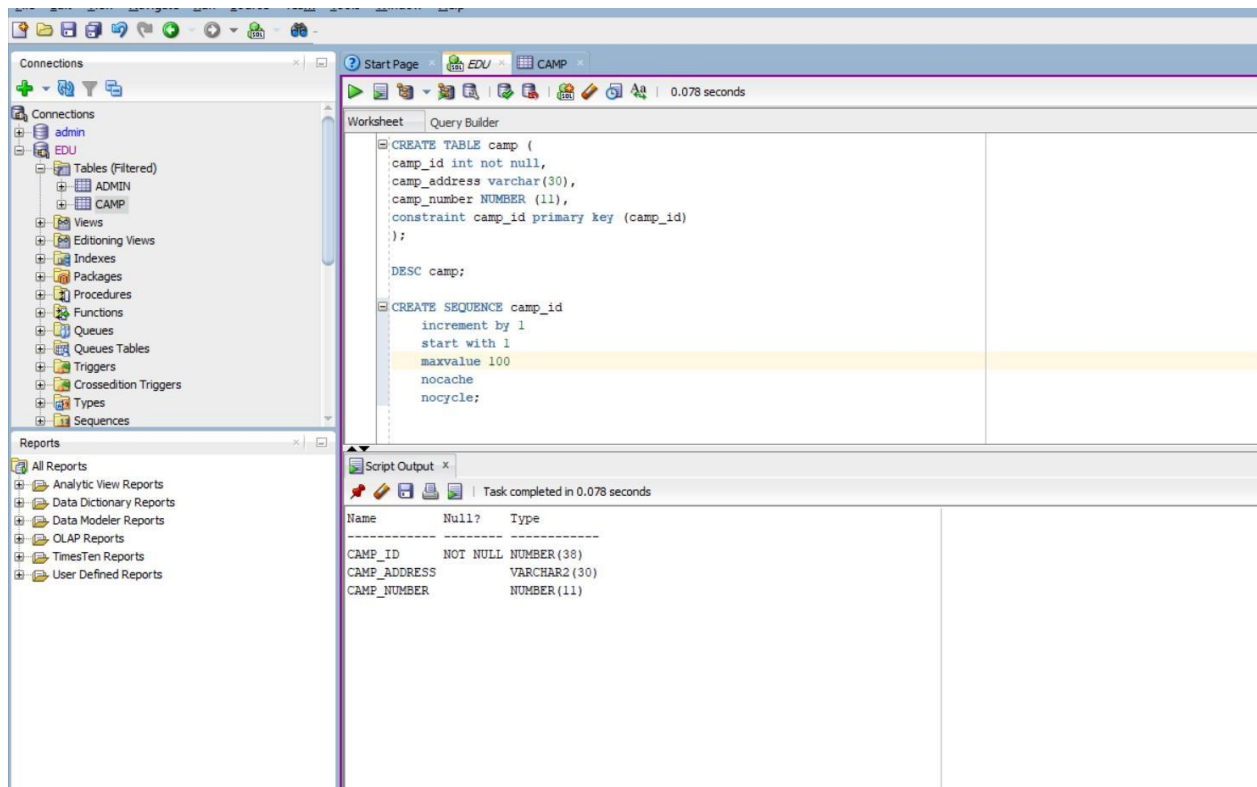
CREATE TABLE camp ( camp_id
int not null, camp_address
varchar(30), camp_number
NUMBER (11),
constraint camp_id primary key (camp_id)
);

```

```

CREATE SEQUENCE camp_id
  increment by 1
start with 1
maxvalue 100
nocache
nocycle;

```



### 3. Employee table:

```

CREATE TABLE employee (
  employee_id int not null,
  admin_id int, camp_id int,
  employee_name varchar(30),
  employee_salary NUMBER(10),
  employee_number NUMBER(11),
  employee_voter_id NUMBER(11),
  employee_job varchar(30),
  employee_joinig_date date,
  employee_age varchar(3),

```

```

employee_gender varchar(5),
employee_adress varchar(30),
constraint employee_id primary key (employee_id),
foreign key (camp_id) references camp (camp_id),
foreign key (admin_id) references admin (admin_id)
);

```

```

CREATE SEQUENCE employee_id
    increment by 1
start with 2
maxvalue 100
nocache
nocycle;

```

Worksheet Query Builder 0.047 seconds

```

CREATE TABLE employee (
  employee_id int not null,
  admin_id int,
  camp_id int,
  employee_name varchar(30),
  employee_salary NUMBER (10),
  employee_number NUMBER (11),
  employee_voter_id NUMBER (11),
  employee_job varchar(30),
  employee_joinig_date date,
  employee_age varchar(3),
  employee_gender varchar(10),
  employee_adress varchar(30),
  constraint employee_id primary key (employee_id),
  foreign key (camp_id) references camp (camp_id),
  foreign key (admin_id) references admin (admin_id)
);

```

Script Output x Task completed in 0.047 seconds

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(38)
ADMIN_ID		NUMBER(38)
CAMP_ID		NUMBER(38)
EMPLOYEE_NAME		VARCHAR2(30)
EMPLOYEE_SALARY		NUMBER(10)
EMPLOYEE_NUMBER		NUMBER(11)
EMPLOYEE_VOTER_ID		NUMBER(11)
EMPLOYEE_JOB		VARCHAR2(30)
EMPLOYEE_JOINIG_DATE		DATE
EMPLOYEE_AGE		VARCHAR2(3)
EMPLOYEE_GENDER		VARCHAR2(10)
EMPLOYEE_ADRESS		VARCHAR2(30)



## 4. Blood table:

```
CREATE TABLE Blood ( blood_id
int not null, blood_group
varchar(4),
primary key (blood_id)
);
```

```
CREATE SEQUENCE blood_id
increment by 1 start with
1 maxvalue 100
nocache
nocycle;
```

The screenshot shows a database query tool interface with two main panes. The top pane, titled 'Query Builder', contains the SQL code for creating a table and a sequence. The bottom pane, titled 'Script Output', shows the execution results.

**Query Builder:**

```
CREATE TABLE Blood (
  blood_id int not null,
  blood_group varchar(4),
  primary key (blood_id)
);
DESC Blood;
CREATE SEQUENCE blood_id
  increment by 1
  start with 1
  maxvalue 100
  nocache
  nocycle;
```

**Script Output:**

Task completed in 0.031 seconds

Table BLOOD created.

Name	Null?	Type
BLOOD_ID	NOT NULL	NUMBER(38)
BLOOD_GROUP		VARCHAR2(4)

Sequence BLOOD\_ID created.

## 5. Doner table:

```
CREATE TABLE doner (  
  doner_id int not null, camp_id  
  int, employee_id int,  
  doner_name varchar(30),  
  blood_id int, doner_number  
  NUMBER (11), doner_address  
  varchar(30), doner_voter_id  
  NUMBER (11), doner_age  
  varchar(3), doner_gender  
  varchar(5),  
  primary key (doner_id),  
  
  FOREIGN KEY (employee_id)  
  REFERENCES employee(employee_id),  
  
  FOREIGN KEY (blood_id)  
  REFERENCES Blood(blood_id),  
  
  FOREIGN KEY (camp_id)  
  REFERENCES camp(camp_id)  
  );  
  
CREATE SEQUENCE doner_id  
  increment by 1  start with  
  1  maxvalue 100  
  nocache  nocycle;
```



## 6. Stock table:

```
CREATE TABLE stock (  
  stock_id int not null,  
  doner_id int, stock_date  
  date, stock_quantity  
  NUMBER,  
  primary key (stock_id),  
  
  FOREIGN KEY (doner_id)  
  REFERENCES doner (doner_id)  
);
```

```
CREATE SEQUENCE stock_id  
  increment by 1  start with  
  1  maxvalue 100  
  nocache  
  nocycle;
```

The screenshot displays a database management interface with a 'Query Builder' tab. The SQL editor contains the following queries:

```

CREATE TABLE stock (
  stock_id int not null,
  doner_id int,
  stock_date date,
  stock_quantity NUMBER,
  primary key (stock_id),
  FOREIGN KEY (doner_id)
  REFERENCES doner (doner_id)
);

DESC stock;

CREATE SEQUENCE stock_id
  increment by 1
  start with 1
  maxvalue 100
  nocache
  nocycle;

```

The 'Script Output' pane shows the results of the execution:

```

Table STOCK created.

Name          Null?   Type
-----
STOCK_ID      NOT NULL  NUMBER(38)
DONER_ID      NUMBER(38)
STOCK_DATE    DATE
STOCK_QUANTITY  NUMBER

Sequence STOCK_ID created.

```

## 7. Secker table:

```

CREATE TABLE secker (
  secker_id int not null,
  secker_name varchar(30),
  secker_age varchar(3),
  secker_voter_id NUMBER (11),
  secker_number NUMBER (11),
  blood_id int, secker_adress
  varchar(30), secker_gender
  varchar(5), primary key
  (secker_id), FOREIGN KEY
  (blood_id)
  REFERENCES Blood(blood_id)
);

CREATE SEQUENCE secker_id
  increment by 1
  start with 1
  maxvalue 100
  nocache  nocycle;

```

The screenshot displays an SQL IDE interface. The main window shows the following SQL script:

```

CREATE TABLE secker (
  secker_id int not null,
  secker_name varchar(30),
  secker_age varchar(3),
  secker_voter_id NUMBER (11),
  secker_number NUMBER (11),
  blood_id int,
  secker_address varchar(30),
  secker_gender varchar(10),
  primary key (secker_id),
  FOREIGN KEY (blood_id)
  REFERENCES Blood(blood_id)
);

DESC secker;

CREATE SEQUENCE secker_id
  increment by 1
  start with 1
  maxvalue 100
  nocache
  nocycle;

```

The 'Script Output' window at the bottom shows the execution results:

Name	Null?	Type
SECKER_ID	NOT NULL	NUMBER(38)
SECKER_NAME		VARCHAR2(30)
SECKER_AGE		VARCHAR2(3)
SECKER_VOTER_ID		NUMBER(11)
SECKER_NUMBER		NUMBER(11)
BLOOD_ID		NUMBER(38)
SECKER_ADDRESS		VARCHAR2(30)
SECKER_GENDER		VARCHAR2(10)

## 8. Hospital table:

```

CREATE TABLE hospital ( hospital_id
int not null, hospital_name
varchar(30),
hospital_number NUMBER (11),
secker_id int,
primary key (hospital_id),

```

```

FOREIGN KEY (secker_id)
REFERENCES secker (secker_id)
);
CREATE SEQUENCE hospital_id
    increment by 1
start with 1
maxvalue 100
nocache
nocycle;

```

The screenshot shows a database query tool interface. The top toolbar indicates the task was completed in 0.032 seconds. The main window is titled 'Worksheet' and 'Query Builder'. It contains the following SQL code:

```

CREATE TABLE hospital (
    hospital_id int not null,
    hospital_name varchar(30),
    hospital_number NUMBER (11),
    secker_id int,
    primary key (hospital_id),
    FOREIGN KEY (secker_id)
    REFERENCES secker (secker_id)
);

DESC secker;

CREATE SEQUENCE hospital_id
    increment by 1
start with 1
maxvalue 100
nocache
nocycle;

```

Below the code, the 'Script Output' window shows the results of the execution:

Table HOSPITAL created.

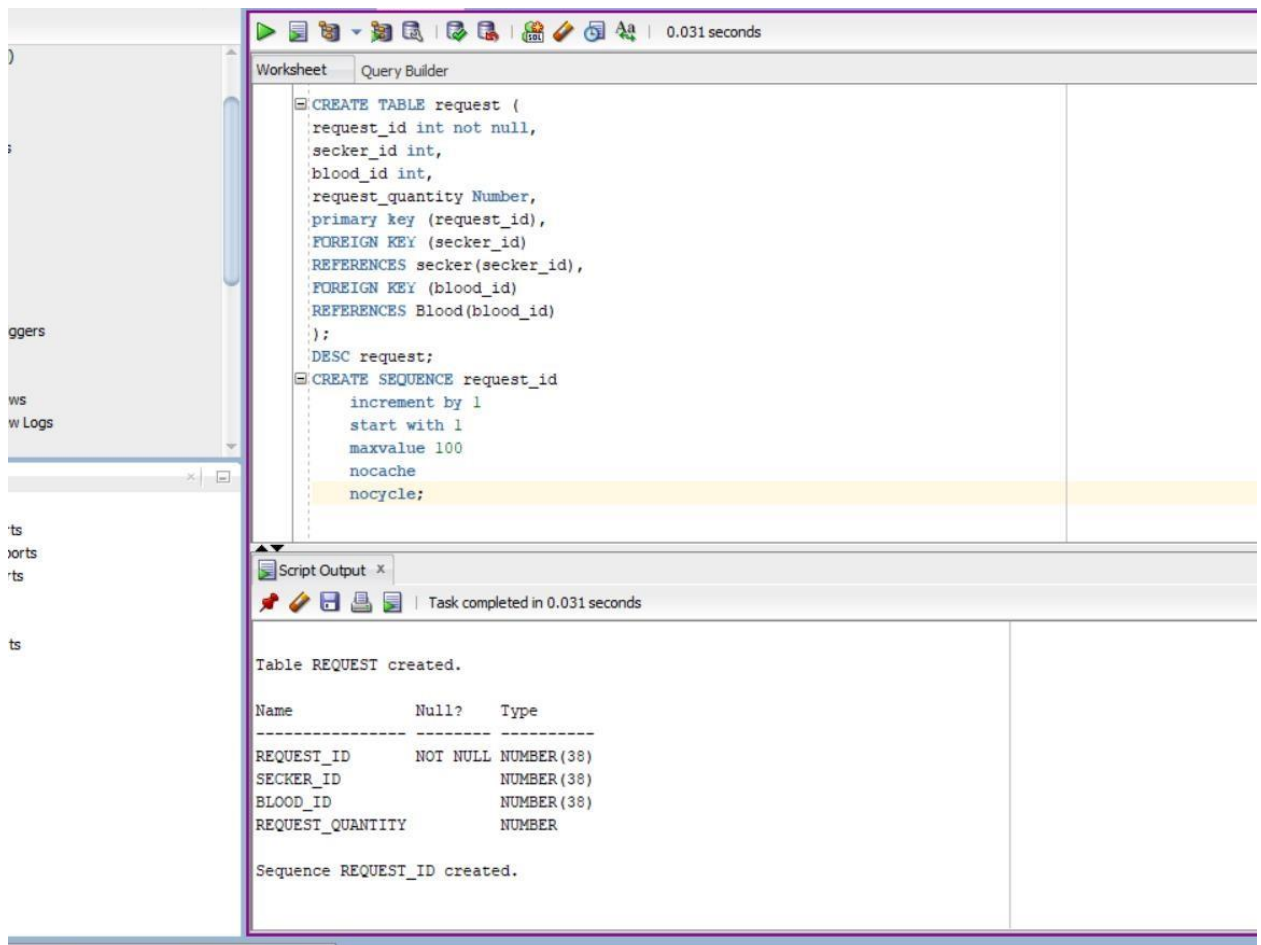
Name	Null?	Type
SECKER_ID	NOT NULL	NUMBER(38)
SECKER_NAME		VARCHAR2(30)
SECKER_AGE		VARCHAR2(3)
SECKER_VOTER_ID		NUMBER(11)
SECKER_NUMBER		NUMBER(11)
BLOOD_ID		NUMBER(38)
SECKER_ADRESS		VARCHAR2(30)
SECKER_GENDER		VARCHAR2(5)

Sequence HOSPITAL\_ID created.

## 9. Recquest table:

```
CREATE TABLE request (  
  request_id int not null,  
  secker_id int, blood_id  
  int, request_quantity  
  Number,  
  primary key (request_id),  
  
  FOREIGN KEY (secker_id)  
  REFERENCES secker(secker_id),  
  
  FOREIGN KEY (blood_id)  
  REFERENCES Blood(blood_id)  
  );  
  
CREATE SEQUENCE request_id  
  increment by 1   start with 1  
  maxvalue 100   nocache  
  nocycle;
```





## 10. Issue table:

```

CREATE TABLE issue (
  issue_id int not null,
  employee_id int,
  request_id int, primary
  key (issue_id),

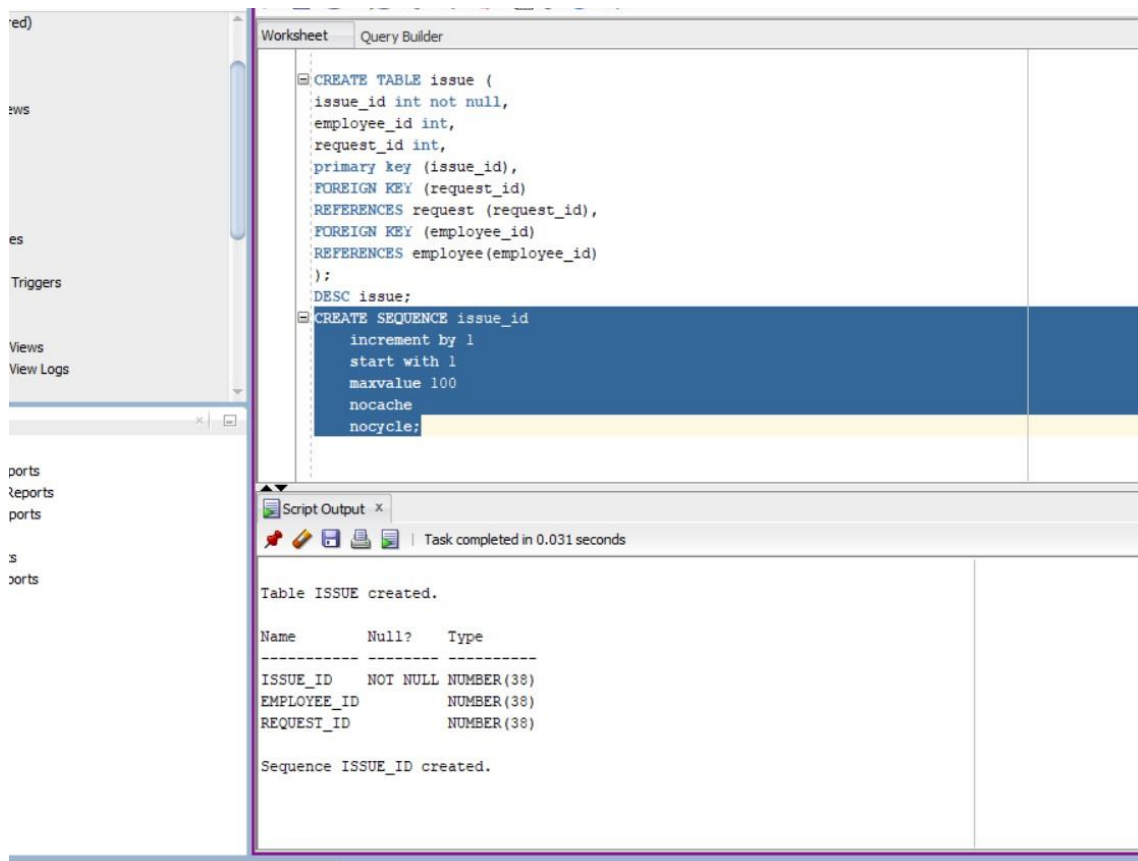
```

```

  FOREIGN KEY (request id)
  REFERENCES request (request_id)
  FOREIGN KEY (employee_id)
  REFERENCES employee(employee_id)
);

```

```
CREATE SEQUENCE issue_id  
increment by 1 start with  
1 maxvalue 100  
nocache nocycle;
```



## Data insertion:

### 1. For Admin table:

insert into admin values (admin\_id.nextval , 'Sakib',  
'sakib10@gmail.com', 'Bashundhora', 121102349,  
21,'male', 01834356432);

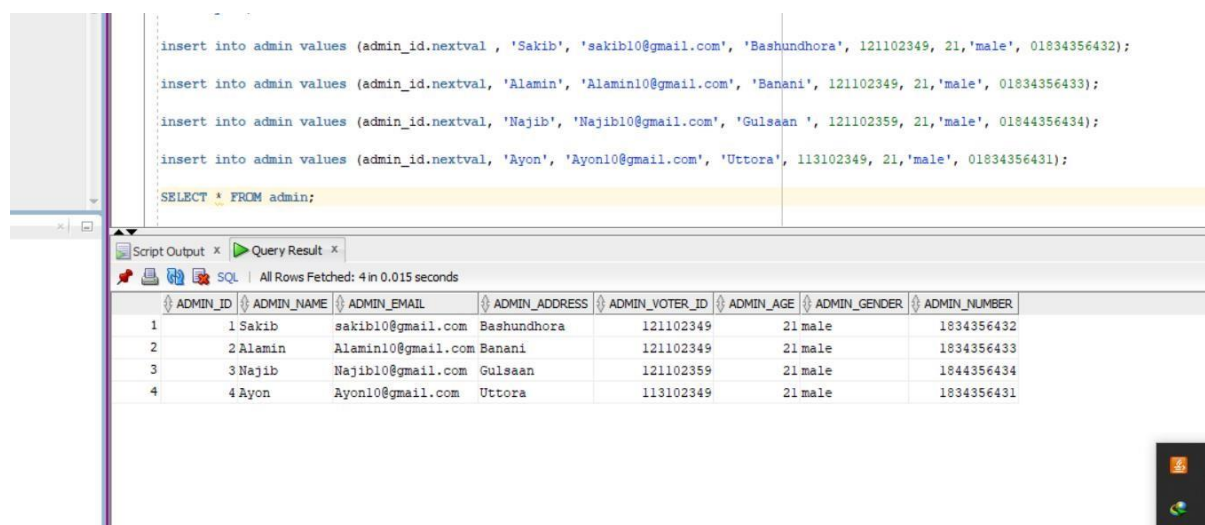
insert into admin values (admin\_id.nextval, 'Amin',  
'Amin10@gmail.com', 'Banani', 121102349, 21,'male',  
01834356433);

insert into admin values (admin\_id.nextval, 'Najib',

```
'Najib10@gmail.com', 'Gulsaan ', 121102359, 21,'male',  
01844356434);
```

```
insert into admin values (admin_id.nextval, 'Ayon',  
'Ayon10@gmail.com', 'Uttora', 113102349, 21,'male',  
01834356431);
```

```
SELECT * FROM admin;
```



The screenshot displays a SQL IDE interface. The top pane shows the following SQL script:

```
insert into admin values (admin_id.nextval , 'Sakib', 'sakib10@gmail.com', 'Bashundhora', 121102349, 21,'male', 01834356432);  
insert into admin values (admin_id.nextval, 'Alamin', 'Alamin10@gmail.com', 'Banani', 121102349, 21,'male', 01834356433);  
insert into admin values (admin_id.nextval, 'Najib', 'Najib10@gmail.com', 'Gulsaan ', 121102359, 21,'male', 01844356434);  
insert into admin values (admin_id.nextval, 'Ayon', 'Ayon10@gmail.com', 'Uttora', 113102349, 21,'male', 01834356431);  
SELECT * FROM admin;
```

The bottom pane shows the 'Query Result' tab with the following data:

ADMIN_ID	ADMIN_NAME	ADMIN_EMAIL	ADMIN_ADDRESS	ADMIN_VOTER_ID	ADMIN_AGE	ADMIN_GENDER	ADMIN_NUMBER
1	Sakib	sakib10@gmail.com	Bashundhora	121102349	21	male	1834356432
2	Alamin	Alamin10@gmail.com	Banani	121102349	21	male	1834356433
3	Najib	Najib10@gmail.com	Gulsaan	121102359	21	male	1844356434
4	Ayon	Ayon10@gmail.com	Uttora	113102349	21	male	1834356431

## 2. Camp table:

```
insert into camp values (camp_id.nextval, 'Banani',  
01934356433);
```

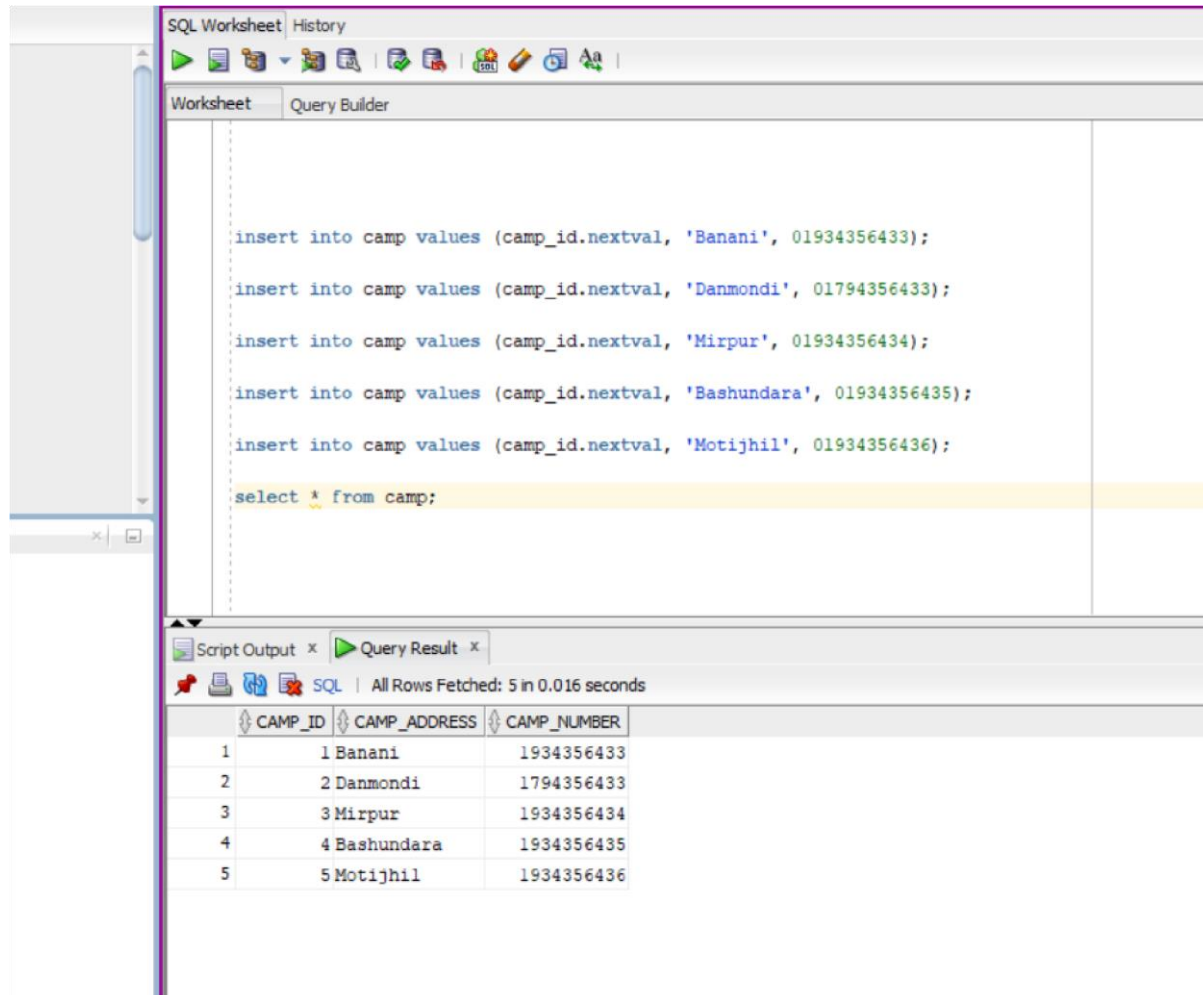
```
insert into camp values (camp_id.nextval, 'Danmondi',  
01794356433);
```

```
insert into camp values (camp_id.nextval, 'Mirpur',  
01934356434);
```

```
insert into camp values (camp_id.nextval, 'Bashundara',  
01934356435);
```

insert into camp values (camp\_id.nextval, 'Motijhil',  
01934356436);

select \* from camp;



The screenshot shows an SQL Worksheet interface. The main area contains the following SQL statements:

```
insert into camp values (camp_id.nextval, 'Banani', 01934356433);  
insert into camp values (camp_id.nextval, 'Danmondi', 01794356433);  
insert into camp values (camp_id.nextval, 'Mirpur', 01934356434);  
insert into camp values (camp_id.nextval, 'Bashundara', 01934356435);  
insert into camp values (camp_id.nextval, 'Motijhil', 01934356436);  
select * from camp;
```

Below the SQL statements, the 'Query Result' tab is active, showing the results of the last query. The status bar indicates 'All Rows Fetched: 5 in 0.016 seconds'. The results are displayed in a table with three columns: CAMP\_ID, CAMP\_ADDRESS, and CAMP\_NUMBER.

CAMP_ID	CAMP_ADDRESS	CAMP_NUMBER
1	1 Banani	1934356433
2	2 Danmondi	1794356433
3	3 Mirpur	1934356434
4	4 Bashundara	1934356435
5	5 Motijhil	1934356436

### 3. Employee table:

insert into employee values

(employee\_id.nextval,5,1,'Najib Hossain',25000

```
,01725352122,91197346751,'Mannager',to_date('22MAR  
CH-2021', 'DD-MM-YYYY'),20,'male','Banani'); insert into  
employee values (employee_id.nextval,6,1,'  
Ayon Ghos',30000  
,01725352123,91827356751,'Cashier',to_date('25-  
April2021', 'DD-MM-YYYY'),22,'male','Banani');
```

```
insert into employee values  
(employee_id.nextval,6,1,'Alamin  
Hossain',25000,01725352132  
,91827846751,'Cashier2',to_date('22-MARCH-2021',  
'DDMM-YYYY'),33,'male','Banani');
```

```
insert into employee values  
(employee_id.nextval,6,1,'Sakibur  
Rahman',25000,01825352122,91827566751,'Cashier3',to  
_date('27-January-2021', 'DD-  
MMYYYY'),44,'male','Banani');
```

```
insert into employee values  
(employee_id.nextval,6,1,'Evan Lews',35000  
,01525352122,91827324751,'Salesman',to_date('22MARC  
H-2021', 'DD-MM-YYYY'),55,'male','Banani');
```

```
insert into employee values  
(employee_id.nextval,6,1,'Flacher',30000,01525352322,9  
1827696751,'Salesman',to_date('21-MARCH-2022',  
'DDMM-YYYY'),55,'male','Banani');
```

```
insert into employee values
(employee_id.nextval,6,2,'Koholi
Saha',35000,01625352122,918273490751,'Salesman',to_
date('12-July-2022', 'DD-MM-YYYY'),33,'male','Banani');
```

```
insert into employee values
(employee_id.nextval,6,2,'Rhohit shrorma ',35000,
01625352133,91878346751,'Salesman ',to_date('12-
July2019', 'DD-MM-YYYY'),33,'male','Banani');
```

```
insert into employee values
(employee_id.nextval,7,2,'Sakibur
Rahman',35000,01525352122,91828346751,'Mannager',t
o_date('24-June-2020', 'DD-MMYYYY'),33,'male','Mirpur');
```

```
insert into employee values
(employee_id.nextval,8,2,'Alamin
Hossain',33000,01525352122,91827347451
,'Mannager',to_date('24-June-2020', 'DD-
MMYYYY'),33,'male','Mirpur');
```

```
insert into employee values
(employee_id.nextval,7,2,'Ayon Ghosh',3500
,01525352122,91821546751,'Mannager',to_date('24Marc
h-2021', 'DD-MM-YYYY'),25,'male','Mirpur');
```

```
insert into employee values
(employee_id.nextval,8,2,'Najib Alam',3000
,01925352122,91863346751,'Accountant',to_date('24-
```

```
November-2020', 'DD-MM-YYYY'),25,'male','Mirpur');  
insert into employee values  
(employee_id.nextval,7,3,'Evan been',35000  
,01525352442,91824346751,'Accountant',to_date('31Nov  
ember-2020', 'DD-MM-YYYY'),25,'male','Mirpur');
```

```
insert into employee values  
(employee_id.nextval,7,3,'Alex  
curray',35000,01525352122,91856346751  
, 'Accountant',to_date('24-January-2020', 'DD-  
MMYYYY'),25,'male','Mirpur');
```

```
insert into employee values (employee_id.nextval,7,3,'Mr  
been',35000,01321352122,91827349851  
, 'Accountant',to_date('20-June-2020', 'DD-  
MMYYYY'),25,'male','Gulshan');
```

```
insert into employee values  
(employee_id.nextval,8,3,'David  
Silva',35000,01443352122,98827346751,'Accountant',to_  
date('24-June-2020', 'DD-MM-YYYY'),25,'male','Gulshan');
```

```
insert into employee values  
(employee_id.nextval,8,4,'Kedar Jadav',32000  
,01525352232,91227346751,'Nurse',to_date('04-  
October2019', 'DD-MM-YYYY'),25,'male','Gulshan');
```

```
insert into employee values  
(employee_id.nextval,8,4,'Ariana
```



```
Alexa',32000,01665352232,91827377751,'Nurse',to_date('04-October-2019', 'DD-MM-YYYY'),28,'female','Gulshan');
```

```
insert into employee values  
(employee_id.nextval,8,4,'Virat  
Kholi',32000,01445352232,91827346541,'Nurse',to_date('04-October-2019', 'DD-MM-YYYY'),28,'male','Gulshan');
```

```
insert into employee values  
(employee_id.nextval,7,5,'Alan  
Walker',32000,01525352277,91827342351  
, 'Nurse',to_date('04-October-2019', 'DD-MM-YYYY'),28,'male','Mirpur');
```

```
insert into employee values  
(employee_id.nextval,7,5,'Cristofar  
Nolan',31000,01746382593,91827342351  
, 'Nurse',to_date('04-January-2021', 'DD-MM-YYYY'),28,'female','Mirpur');
```

```
insert into employee values  
(employee_id.nextval,8,5,'Katrina  
Kayef',30000,01525344892,91823346751  
, 'Nurse',to_date('12-January-2020', 'DD-MM-YYYY'),28,'female','Mirpur');
```

```
select * from employee;
```

EMPLOYEE_ID	ADMIN_ID	CAMP_ID	EMPLOYEE_NAME	EMPLOYEE_SALARY	EMPLOYEE_NUMBER	EMPLOYEE_VOTER_ID	EMPLOYEE_JOB	EMPLOYEE_JOINING_DATE	EMPLOYEE_AGE	EMPLOYEE_GENDER	EMPLOYEE_ADR
1	2	5	1 Najib Hossain	25000	1725352122	91827346751	Manager	22-MAR-21	20	male	Banani
2	3	6	1 Ayon Ghos	30000	1725352123	91827356751	Cashier	25-APR-21	22	male	Banani
3	4	6	1 Alamin Hossain	25000	1725352132	91827846751	Cashier2	22-MAR-21	33	male	Banani
4	5	6	1 Sakibur Rahman	25000	1825352122	91827566751	Cashier3	27-JAN-21	44	male	Banani
5	6	6	1 Evan Lewis	35000	1525352122	91827324751	Salesman	22-MAR-21	55	male	Banani
6	7	6	1 Flacher	30000	1525352322	91827696751	Salesman	21-MAR-22	55	male	Banani
7	9	6	2 Rhohit shorma	35000	1625352133	91878346751	Salesman	12-JUL-19	33	male	Banani
8	10	7	2 Sakibur Rahman	35000	1525352122	91828346751	Manager	24-JUN-20	33	male	Mirpur
9	11	8	2 Alamin Hossain	33000	1525352122	91827347451	Manager	24-JUN-20	33	male	Mirpur
10	12	7	2 Ayon Ghosh	3500	1525352122	91821546751	Manager	24-MAR-21	25	male	Mirpur
11	13	8	2 Najib Alam	3000	1925352122	91863346751	Accountant	24-NOV-20	25	male	Mirpur
12	15	7	3 Alex curray	35000	1525352122	91856346751	Accountant	24-JAN-20	25	male	Mirpur
13	16	7	3 Mr been	35000	1321352122	91827349851	Accountant	20-JUN-20	25	male	Gulshan
14	17	8	3 David Silve	35000	1443352122	98827346751	Accountant	24-JUN-20	25	male	Gulshan
15	18	8	4 Kedar Jadhav	32000	1525352232	91227346751	Nurse	04-OCT-19	25	male	Gulshan
16	19	8	4 Ariana Alexa	32000	1665352232	91827377751	Nurse	04-OCT-19	28	female	Gulshan
17	20	8	4 Virat Kholi	32000	1445352232	91827346541	Nurse	04-OCT-19	28	male	Gulshan
18	21	7	5 Alan Walker	32000	1525352277	91827342351	Nurse	04-OCT-19	28	male	Mirpur
19	22	7	5 Cristofar Nolan	31000	1746382593	91827342351	Nurse	04-JAN-21	28	female	Mirpur
20	23	8	5 Katrina Kayef	30000	1525344892	91823346751	Nurse	12-JAN-20	28	female	Mirpur

#### 4. Blood table:

insert into Blood values (blood\_id.nextval,'b+');

insert into Blood values (blood\_id.nextval,'o+');

insert into Blood values (blood\_id.nextval,'ab-');

insert into Blood values (blood\_id.nextval,'a-');

insert into Blood values (blood\_id.nextval,'b-');

insert into Blood values (blood\_id.nextval,'o-');

select \* from Blood;

BLOOD_ID	BLOOD_GROUP
1	1 ab+
2	2 a+
3	3 b+
4	4 o+
5	5 ab-
6	6 a-
7	7 b-
8	8 o-

## 5. Doner table:

insert into doner values

```
(doner_id.nextval,1,21,'Sakib',1,018344763,'Banani',21343455,21,'male'); insert
```

into doner values

```
(doner_id.nextval,2,19,'Saimon',2,018343763,'Gulshan',21343456,23,'male'); insert
```

into doner values

```
(doner_id.nextval,3,19,'Hridoy',1,018347633,'Bashundara',21343457,21,'male'); insert
```

into doner values

```
(doner_id.nextval,2,22,'Adnan',4,018347632,'Mirpur',21343458,27,'male'); insert
```

into doner values

```
(doner_id.nextval,3,23,'Udoy',6,018347217,'Shahbag',21343459,29,'male');
```

insert into doner values

```
(doner_id.nextval,5,18,'Alexa',4,018342139,'Airport',21343453,25,'female'); insert
```

into doner values

```
(doner_id.nextval,4,18,'JON',4,01834542139,'Airport',2134453453,25,'male'); insert
```

into doner values

```
(doner_id.nextval,3,19,'RICK',4,01849342139,'UTTARA',2134373453,24,'MALE'); insert
```

into doner values

```
(doner_id.nextval,3,19,'FLEX',4,018342101639,'MOTIZIL',2
```

```

1343436874,23,'MALE'); insert
into doner values
(doner_id.nextval,4,21,'JACK',4,018344569739,'DHONMO
NDI',2133043453,21,'MALE'); insert into doner values
(doner_id.nextval,5,20,'ANDERSON',4,01037698139,'AIUB
',2145683434,25,'MALE'); insert
into doner values
(doner_id.nextval,1,22,'NAJIB',4,018342139734,'UTTARA',
21343453123,24,'MALE'); insert
into doner values
(doner_id.nextval,5,23,'HOSSAIN',4,01834242139,'DIABAR
I',213434216784,26,'MALE');
select * from doner;

```

## 6. Stock table:

```

insert into stock values
(stock_id.nextval,1,to_date('04may-2019', 'DD-MM-YYYY'),1);
insert into stock values (stock_id.nextval,2,to_date('04-january-
2019', 'DD-MM-YYYY'),1);
insert into stock values (stock_id.nextval,3,to_date('04-june-2019',
'DD-MM-YYYY'),1);

```

Filter:										
	DONER_ID	CAMP_ID	EMPLOYEE_ID	DONER_NAME	BLOOD_ID	DONER_NUMBER	DONER_ADDRESS	DONER_VOTER_ID	DONER_AGE	DONER_GENDER
1	1	1	21 Sakib		1	18344763 Banani		21343455	21	male
2	2	2	19 Saimon		2	18343763 Gulshan		21343456	23	male
3	3	3	19 Hridoy		1	18347633 Bashundara		21343457	21	male
4	4	2	22 Adnan		4	18347632 Mirpur		21343458	27	male
5	5	3	23 Uday		6	18347217 Shahbag		21343459	29	male
6	6	5	18 Alexa		4	18342139 Airport		21343453	25	female
7	7	4	18 JON		4	1834542139 Airport		2134453453	25	male
8	8	3	19 RICK		4	1849342139 UTTARA		2134373453	24	MALE
9	14	3	19 RICK hossen		4	1849342139 UTTARA		2134373453	24	MALE
10	15	3	19 FLEX evan		4	18342101639 MOTIZIL		2134343687	23	MALE
11	20	3	19 RICK sikdar		4	1849342139 UTTARA		2134373453	24	MALE
12	21	3	19 FLEX		4	18342101639 MOTIZIL		2134343687	23	MALE
13	22	4	21 JACK		4	18344569739 DHONMONDI		2133043453	21	MALE
14	23	5	20 ANDERSON		4	1037698139 AIUB		2145683434	25	MALE
15	24	1	22 NAJIB		4	18342139734 UTTARA		21343453123	24	MALE

insert into stock values (stock\_id.nextval,4,to\_date('04-june-2019',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,5,to\_date('04-june-2019',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,7,to\_date('04-may-2019',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,6,to\_date('04-july-2019',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,1,to\_date('04-October2020',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,1,to\_date('04-may-2020',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,2,to\_date('04-january-  
2020', 'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,3,to\_date('04-june-2020',  
'DD-MM-YYYY'),1);

insert into stock values (stock\_id.nextval,4,to\_date('04-june-2020',  
'DD-MM-YYYY'),1);

```
insert into stock values (stock_id.nextval,5,to_date('04-june-2020',  
'DD-MM-YYYY'),1);
```

```
insert into stock values (stock_id.nextval,7,to_date('04-may-2020',  
'DD-MM-YYYY'),1);
```

```
insert into stock values (stock_id.nextval,6,to_date('04-july-2020',  
'DD-MM-YYYY'),1);
```

```
insert into stock values (stock_id.nextval,1,to_date('04-October-  
2022', 'DD-MM-YYYY'),1);
```

```
insert into stock values (stock_id.nextval,14,to_date('04-july-2019',  
'DD-MM-YYYY'),1);
```

```
select * from stock;
```

	STOCK_ID	DONER_ID	STOCK_DATE	STOCK_QUANTITY
1	1	1	04-MAY-19	1
2	2	2	04-JAN-19	1
3	3	3	04-JUN-19	1
4	4	4	04-JUN-19	1
5	5	5	04-JUN-19	1
6	6	6	04-MAY-19	1
7	7	7	04-JUL-19	1
8	8	8	04-OCT-20	1
9	9	9	04-JAN-20	1
10	10	10	04-JUN-20	1
11	11	11	04-JUN-20	1
12	12	12	04-JUN-20	1
13	13	13	04-MAY-20	1
14	14	14	04-JUL-20	1
15	15	15	04-OCT-22	1
16	16	16	04-JUL-19	1

## 7. Secker table:

```

INSERT INTO secker VALUES (secker_id.nextval, 'Tom', 21,
2864899071, 012144590,1,'Uttara','male');
INSERT INTO secker VALUES (secker_id.nextval, 'Monty',
21, 2864894472, 012144340,2,'Airport','male');
INSERT INTO secker VALUES (secker_id.nextval, 'Alexa',
21, 2864877071, 012144590,3,'Mirpur','female');
INSERT INTO secker VALUES (secker_id.nextval, 'Ariana',
20, 2554899077, 014544590,3,'Azimpur','female');
INSERT INTO secker VALUES (secker_id.nextval, 'warner',
26, 2864899073, 012144593,2,'Gulshan','male');

```

```

INSERT INTO secker VALUES (secker_id.nextval, 'Willson',
28, 2864899091, 012144589,5,'Mohammadpur','male');
INSERT INTO secker VALUES (secker_id.nextval, 'kazi', 21,
28648990691, 01215456989,5,'Mohammadpur','male');
INSERT INTO secker VALUES (secker_id.nextval, 'ALADIN',
24, 286469091, 01214855214,5,'KHILKHET','male');
INSERT INTO secker VALUES (secker_id.nextval, 'Akin', 23,
2864899042, 01421444459,5,'Mohammadpur','male');
INSERT INTO secker VALUES (secker_id.nextval, 'Onik', 28,
28644568091, 01246131449,5,'Mohammadpur','male');
select * from secker;

```

	SECKER_ID	SECKER_NAME	SECKER_AGE	SECKER_VOTER_ID	SECKER_NUMBER	BLOOD_ID	SECKER_ADRESS	SECKER_GENDER
1	1	Tom	21	2864899071	12144590		1 Uttara	male
2	2	Monty	21	2864894472	12144340		2 Airport	male
3	3	Alexa	21	2864877071	12144590		3 Mirpur	female
4	4	Ariana	20	2554899077	14544590		3 Azimpur	female
5	5	warner	26	2864899073	12144593		2 Gulshan	male
6	6	Willson	28	2864899091	12144589		5 Mohammadpur	male
7	8	kazi	21	28648990691	1215456989		5 Mohammadpur	male
8	9	ALADIN	24	286469091	1214855214		5 KHILKHET	male
9	10	Akin	23	2864899042	1421444459		5 Mohammadpur	male
10	11	Onik	28	28644568091	1246131449		5 Mohammadpur	male

## 8. Hospital table:

```

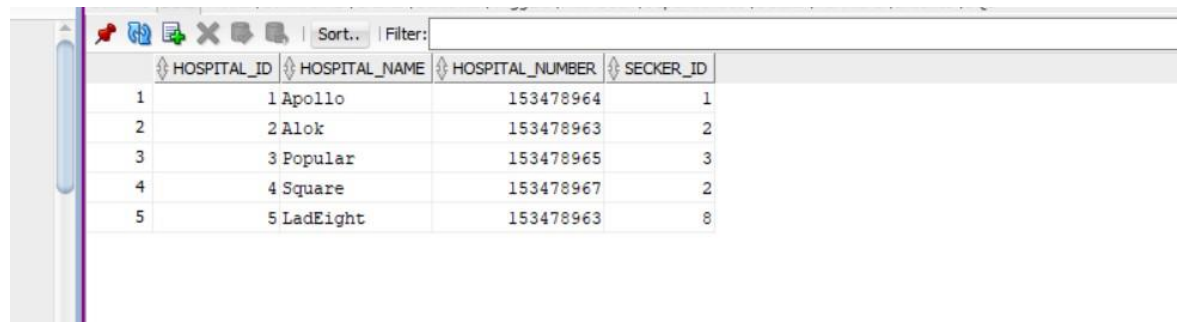
INSERT INTO hospital VALUES (hospital_id.nextval,
'Apollo',0153478964, 1);
INSERT INTO hospital VALUES (hospital_id.nextval,

```



```
'Alok',0153478963, 2);
INSERT INTO hospital VALUES (hospital_id.nextval,
'Popular',0153478965, 3);
INSERT INTO hospital VALUES (hospital_id.nextval,
'Square',0153478967, 2);
INSERT INTO hospital VALUES (hospital_id.nextval,
'LadEight',0153478963, 8);
```

```
select * from hospital;
```



	HOSPITAL_ID	HOSPITAL_NAME	HOSPITAL_NUMBER	SECKER_ID
1	1	Apollo	153478964	1
2	2	Alok	153478963	2
3	3	Popular	153478965	3
4	4	Square	153478967	2
5	5	LadEight	153478963	8

## 9. Recquest table:

```
INSERT INTO request VALUES (request_id.nextval, 1,1,2);
INSERT INTO request VALUES (request_id.nextval, 5,2,1);
INSERT INTO request VALUES (request_id.nextval, 1,4,1);
INSERT INTO request VALUES (request_id.nextval, 6,2,2);
INSERT INTO request VALUES (request_id.nextval, 5,7,1);
INSERT INTO request VALUES (request_id.nextval, 2,3,2);
INSERT INTO request VALUES (request_id.nextval, 6,5,1);
```

```

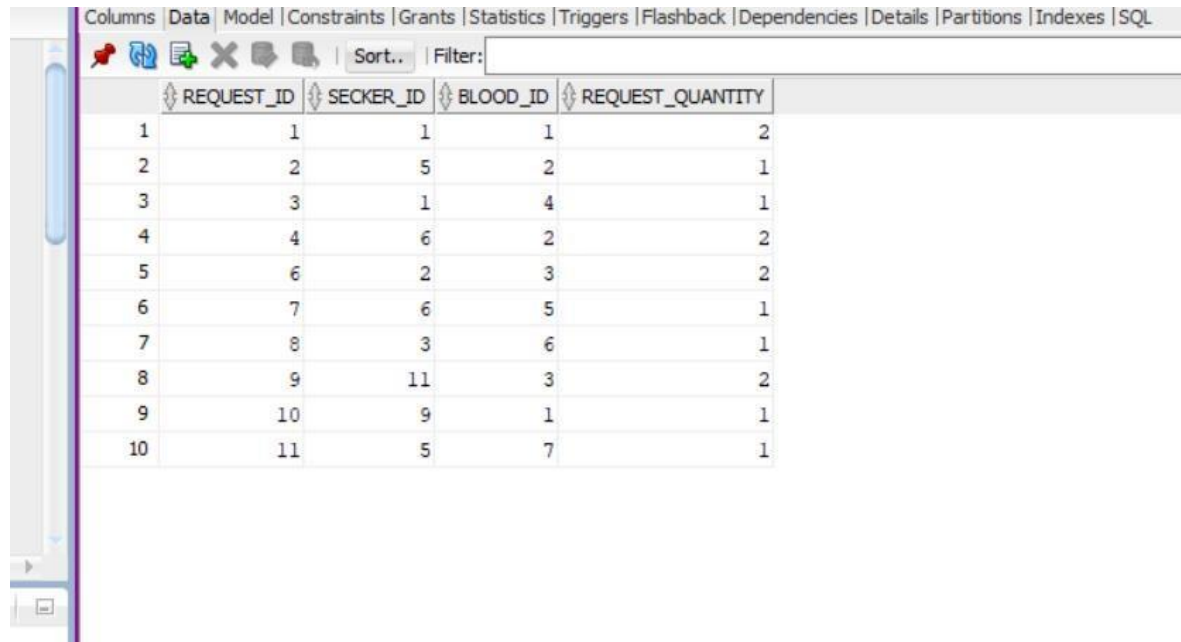
INSERT INTO request VALUES (request_id.nextval, 3,6,1);
INSERT INTO request VALUES (request_id.nextval, 11,3,2);
INSERT INTO request VALUES (request_id.nextval, 9,1,1);

```

```

select * from request;

```



	REQUEST_ID	SECKER_ID	BLOOD_ID	REQUEST_QUANTITY
1	1	1	1	2
2	2	2	5	1
3	3	3	1	4
4	4	4	6	2
5	6	6	2	3
6	7	6	5	1
7	8	3	6	1
8	9	11	3	2
9	10	9	1	1
10	11	5	7	1

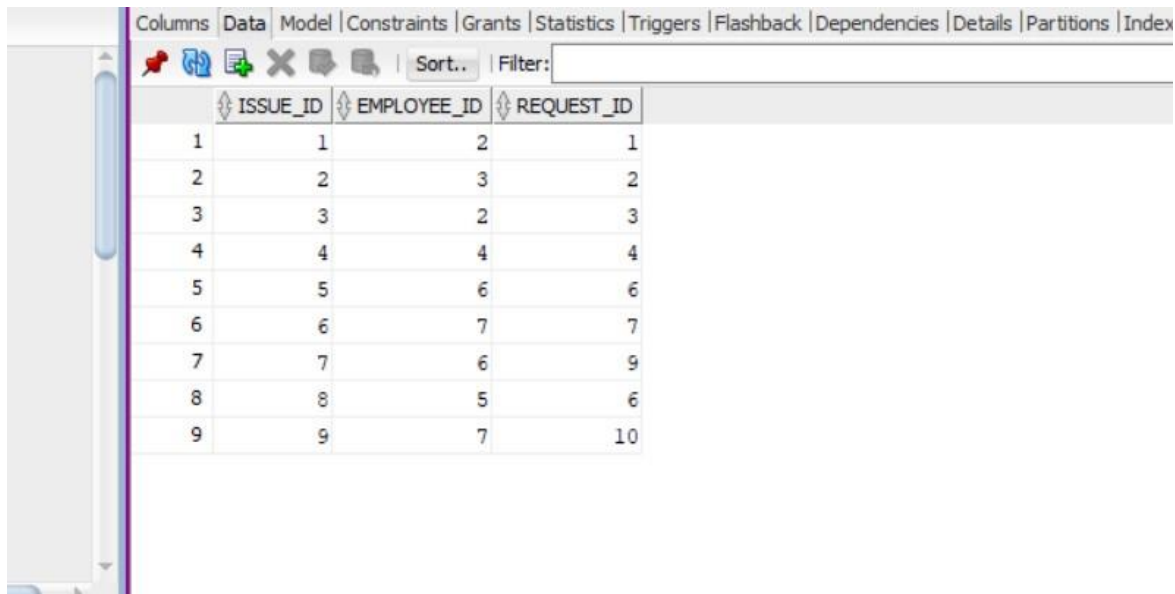
## 10. Issue table:

```

insert into issue values (issue_id.nextval,2,1); insert
into issue values (issue_id.nextval,3,2); insert into
issue values (issue_id.nextval,2,3); insert into issue
values (issue_id.nextval,4,4); insert into issue
values (issue_id.nextval,5,6); insert into issue
values (issue_id.nextval,6,6); insert into issue
values (issue_id.nextval,7,7); insert into issue
values (issue_id.nextval,6,9); insert into issue
values (issue_id.nextval,7,10);

```

select \* from issue;



	ISSUE_ID	EMPLOYEE_ID	REQUEST_ID	
1	1	2	1	
2	2	3	2	
3	3	2	3	
4	4	4	4	
5	5	6	6	
6	6	7	7	
7	7	6	9	
8	8	5	6	
9	9	7	10	

## Query writings based on the tables:

- **Joining:**

### Inner join :

#### 1. Show employee id, doner id, doner name, doner age:

```
Select d.employee_id, d.doner_id,  
d.doner_name, d.doner_age from  
doner d  
join employee e on  
d.employee_id=e.employee_id;
```

tions

Worksheet      Query Builder

```
select d.employee_id, d.doner_id, d.doner_name, d.doner_age
from doner d
join employee e on d.employee_id =e.employee_id;
```

Query Result x

SQL | All Rows Fetched: 15 in 0.031 seconds

	EMPLOYEE_ID	DONER_ID	DONER_NAME	DONER_AGE
1	21	1	Sakib	21
2	19	2	Saimon	23
3	19	3	Hridoy	21
4	22	4	Adnan	27
5	23	5	Udoy	29
6	18	6	Alexa	25
7	18	7	JON	25
8	19	8	RICK	24
9	19	14	RICK hossen	24
10	19	15	FLEX evan	23
11	19	20	RICK sikdar	24
12	19	21	FLEX	23
13	21	22	JACK	21
14	20	23	ANDERSON	25
15	22	24	NAJIB	24

Right join:

2. Show all the employee who has no doner associated with them with right join  
employee id, doner id, doner name,  
doner age:

```
SELECT e.employee_id, d.doner_id,  
d.doner_name, d.doner_age FROM  
doner d  
RIGHT OUTER JOIN employee e  
ON d.employee_id= e.employee_id;
```

```
SELECT e.employee_id, d.doner_id, d.doner_name, d.doner_age
FROM doner d
RIGHT OUTER JOIN employee e
ON d.employee_id= e.employee_id;
```

Query Result x

SQL | All Rows Fetched: 29 in 0 seconds

	EMPLOYEE_ID	DONER_ID	DONER_NAME	DONER_AGE
1	21	1	Sakib	21
2	19	2	Saimon	23
3	19	3	Hridoy	21
4	22	4	Adnan	27
5	23	5	Udoy	29
6	18	6	Alexa	25
7	18	7	JON	25
8	19	8	RICK	24
9	19	14	RICK hossen	24
10	19	15	FLEX evan	23
11	19	20	RICK sikdar	24
12	19	21	FLEX	23
13	21	22	JACK	21
14	20	23	ANDERSON	25
15	22	24	NAJIB	24
16	5	(null)	(null)	(null)
17	3	(null)	(null)	(null)
18	15	(null)	(null)	(null)
19	12	(null)	(null)	(null)
20	10	(null)	(null)	(null)

Messages - 1 on

### Left join:

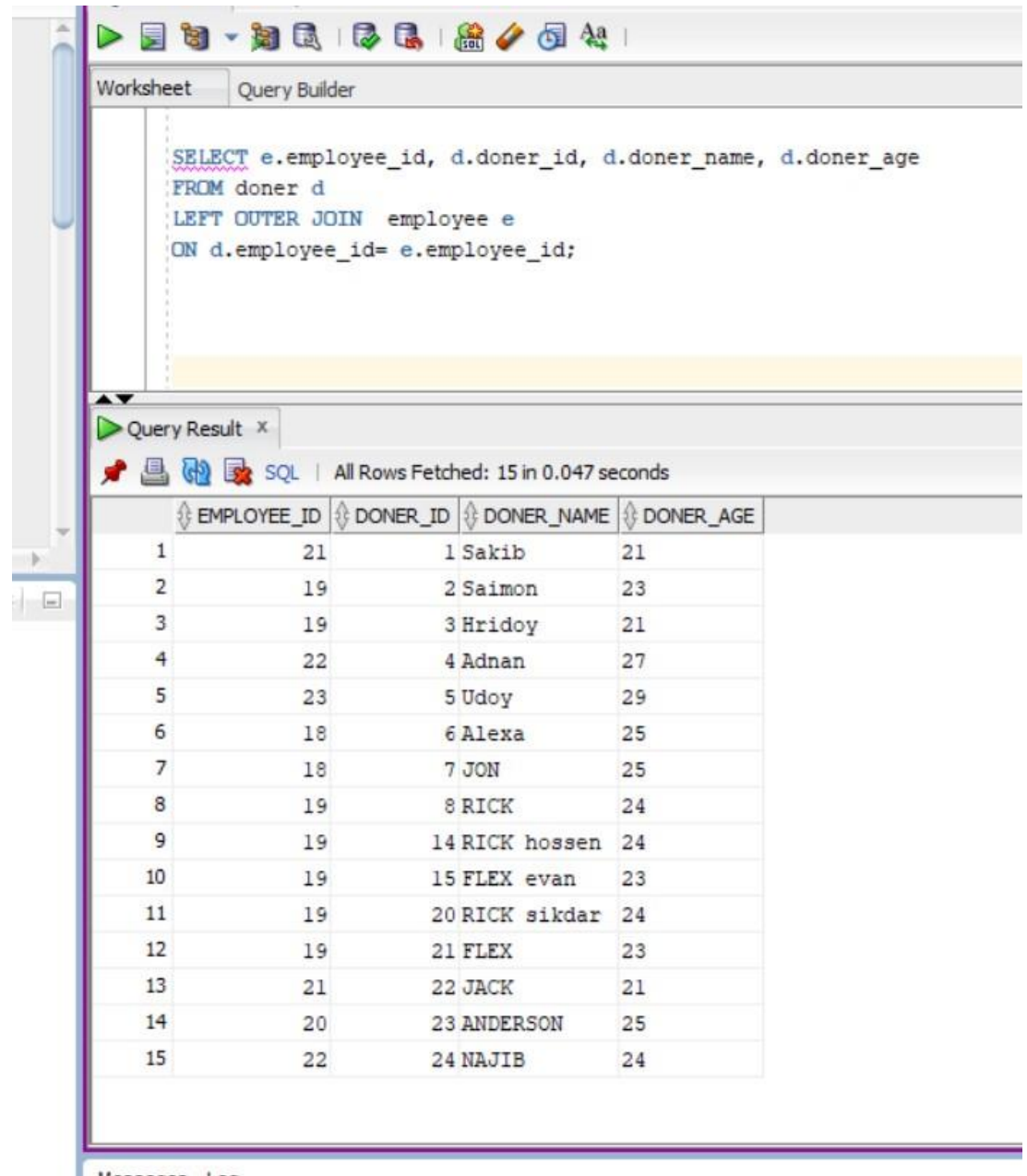
3. Show all the doner who has no employee associated with them with left join  
employee id, doner id, doner name,  
doner age:

```

SELECT e.employee_id, d.doner_id, d.doner_name,
d.doner_age

```

FROM doner d  
LEFT OUTER JOIN employee e  
ON d.employee\_id= e.employee\_id;



The screenshot shows a database query builder interface. The top section is labeled 'Worksheet' and 'Query Builder'. It contains a SQL query:   
`SELECT e.employee_id, d.doner_id, d.doner_name, d.doner_age  
FROM doner d  
LEFT OUTER JOIN employee e  
ON d.employee_id= e.employee_id;`  
The bottom section is labeled 'Query Result' and shows the results of the query. It indicates 'All Rows Fetched: 15 in 0.047 seconds'. The results are displayed in a table with four columns: EMPLOYEE\_ID, DONER\_ID, DONER\_NAME, and DONER\_AGE. The table contains 15 rows of data.

	EMPLOYEE_ID	DONER_ID	DONER_NAME	DONER_AGE
1	21	1	Sakib	21
2	19	2	Saimon	23
3	19	3	Hridoy	21
4	22	4	Adnan	27
5	23	5	Udoy	29
6	18	6	Alexa	25
7	18	7	JON	25
8	19	8	RICK	24
9	19	14	RICK hossen	24
10	19	15	FLEX evan	23
11	19	20	RICK sikdar	24
12	19	21	FLEX	23
13	21	22	JACK	21
14	20	23	ANDERSON	25
15	22	24	NAJIB	24

Full join:

**4. Show all the doner employee associated and not associated with full join employee id, doner id, doner name, doner age:**

```
SELECT e.employee_id, d.doner_id, d.doner_name,  
d.doner_age  
FROM doner d  
FULL OUTER JOIN employee e  
ON d.employee_id= e.employee_id;
```



SQL Worksheet History

Worksheet Query Builder

```

SELECT e.employee_id, d.doner_id, d.doner_name, d.doner_age
FROM doner d
FULL OUTER JOIN employee e
ON d.employee_id= e.employee_id;

```

Query Result x

All Rows Fetched: 29 in 0.005 seconds

	EMPLOYEE_ID	DONER_ID	DONER_NAME	DONER_AGE
1		2	(null) (null)	(null)
2		3	(null) (null)	(null)
3		4	(null) (null)	(null)
4		5	(null) (null)	(null)
5		6	(null) (null)	(null)
6		7	(null) (null)	(null)
7		9	(null) (null)	(null)
8		10	(null) (null)	(null)
9		11	(null) (null)	(null)
10		12	(null) (null)	(null)
11		13	(null) (null)	(null)
12		15	(null) (null)	(null)
13		16	(null) (null)	(null)
14		17	(null) (null)	(null)
15		18	7 JON	25
16		18	6 Alexa	25
17		19	21 FLEX	23
18		19	20 RICK sikdar	24
19		19	15 FLEX evan	23
20		19	14 RICK hossen	24

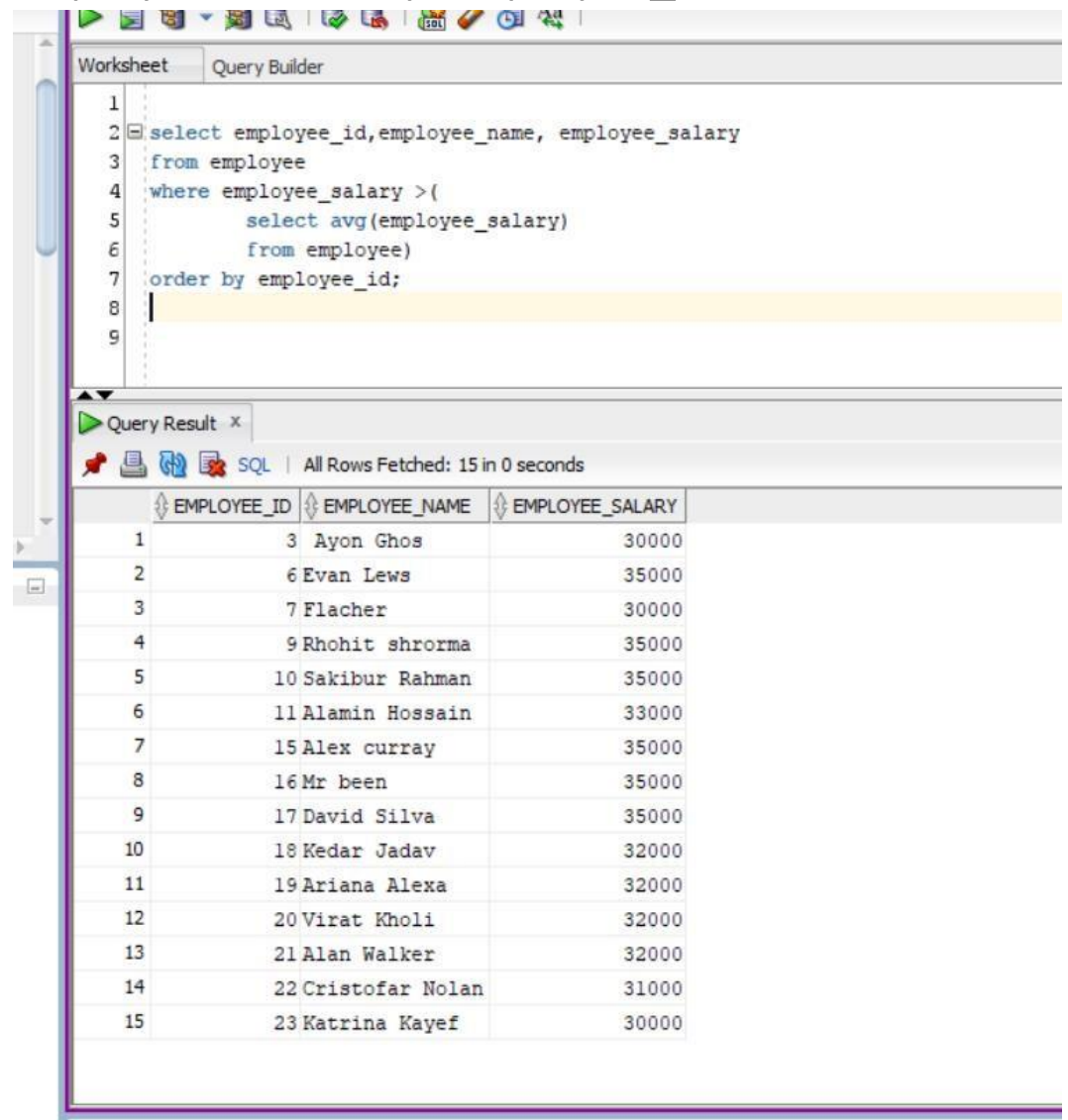
Messages - Log

Messages Statements

- Subquery

## 5. Finding the details of the employee who has more then average:

```
select
employee_id,employee_name,
employee_salary from employee
where employee_salary >(    select
avg(employee_salary)    from
employee) order by employee_id;
```



The screenshot shows a database query builder interface with a 'Query Builder' tab selected. The SQL query is entered in the text area. Below the query, the 'Query Result' window displays the results of the query. The results are shown in a table with three columns: EMPLOYEE\_ID, EMPLOYEE\_NAME, and EMPLOYEE\_SALARY. The table contains 15 rows of data, sorted by EMPLOYEE\_ID in ascending order. The status bar indicates 'All Rows Fetched: 15 in 0 seconds'.

	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_SALARY
1	3	Ayon Ghos	30000
2	6	Evan Lews	35000
3	7	Flacher	30000
4	9	Rhohit shorma	35000
5	10	Sakibur Rahman	35000
6	11	Alamin Hossain	33000
7	15	Alex curray	35000
8	16	Mr been	35000
9	17	David Silva	35000
10	18	Kedar Jadav	32000
11	19	Ariana Alexa	32000
12	20	Virat Kholi	32000
13	21	Alan Walker	32000
14	22	Cristofar Nolan	31000
15	23	Katrina Kayef	30000

- **View**

1. **Creating a view to see the employee details hired by admin\_id 6:**


```
create view employee_hired_by_admin_6 as  
select employee_id, employee_name,  
employee_salary from employee where  
admin_id= 6;
```

```
select * from employee_hired_by_admin_6;
```

## Worksheet Query Builder

```
11  
12  
13 create view employee_hired_by_admin_6 as  
14 select employee_id, employee_name, employee_salary  
15 from employee  
16 where admin_id= 6;  
17  
18  
19 select * from employee_hired_by_admin_6;  
20
```

Script Output x Query Result x

 SQL | All Rows Fetched: 6 in 0 seconds

	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_SALARY
1	3	Ayon Ghos	30000
2	4	Alamin Hossain	25000
3	5	Sakibur Rahman	25000
4	6	Evan Lews	35000
5	7	Flacher	30000
6	9	Rohit shorma	35000