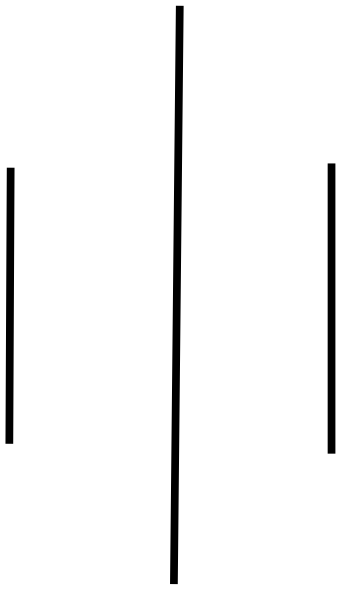
Deerwalk Institute Of Technology
Advance Database Management System



Lab: 3

Submitted By:
Name: Sagar Giri
Roll No. 205

Submitted To:
Rishikesh Katuwal

Date:

Q 1. Create a table which is capable to have multiple values in a field. Insert sample values and display result with one of the values from column having multiple values.

Creating table:

```
CREATE TABLE public.student
(
  sid bigint NOT NULL,
  contact character(10)[],
  name character(50),
  batch character(4),
  address character(80),
  email character(80),
  CONSTRAINT student_pkey PRIMARY KEY (sid)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.student
  OWNER TO postgres;
```

Inserting sample values in "student" table:

```
INSERT INTO student(sid, name, batch, address, contact, email)
VALUES
(201, 'Ashim Regmi', '2016', 'Chabahil',
'{"9807368806","9842076488"}','aregmee@gmail.com'),
(205, 'Sagar Giri', '2016', 'Chabahil',
'{"9845123456","9521456789"}','girisagar46@gmail.com'),
(218, 'Bimal Gaire', '2016', 'Koteshor', '{"9845123456"}','bimalblee@gmail.com');
```

Displaying Output:

```
SELECT * FROM public.student;
```

Output pane

Data Output | Explain | Messages | History

	sid bigint	contact character(10)[]	name character(50)	batch character(4)	address character(80)
1	201	{9807368806,9842076488}	Ashim Regmi	2016	Chabahil
2	205	{9845123456,9521456789}	Sagar Giri	2016	Chabahil
3	218	{9845123456}	Bimal Gaire	2016	Koteshor

Displaying single column output:

```
SELECT sid,name,batch,address,contact[1],email FROM student;
```

Output pane

Data Output | Explain | Messages | History

	sid bigint	name character(50)	batch character(4)	address character(80)	contact character(10)	email character(80)
1	201	Ashim Regmi	2016	Chabahil	9807368806	aregmee@gmail.com
2	205	Sagar Giri	2016	Chabahil	9845123456	girisagar46@gmail.com
3	218	Bimal Gaire	2016	Koteshor	9845123456	bimalblee@gmail.com

2. Create a table using user defined data types. Insert sample values.

To create a table with user defined data types, first we need to define the data types as:

```
CREATE DOMAIN course varchar(10)
CHECK (VALUE IN ('BSCCSIT','BBA','BIT'));

CREATE TYPE batch
AS ENUM ('2016','2017','2018','2019');
```

Now, we create a table using the defined data types:

```
CREATE TABLE customStudent
(
    sid bigint NOT NULL,
    name character(50),
    batch batch,
    course course,
    address character(80),
    contact character(10),
    email character(80),
    CONSTRAINT pk_sid PRIMARY KEY (sid)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE customStudent OWNER TO postgres;
```

Inserting data in our custom table:

```
INSERT INTO customStudent
(sid, name, batch, course, address, contact, email)
VALUES
('201','Asim
Regmi','2016','BIT','Chabahil','9807355656','asim.regmi@deerwalk.edu.np'),
('205','Sagar
Giri','2016','BSCCSIT','Chabahil','9807368806','sagar.giri@deerwalk.edu.np'),
('206','Sameer
Koirala','2016','BSCCSIT','Koteshor','9845123456','sameer.koirala@deerwalk.edu.np')
;
```

Displaying data:

```
SELECT * FROM public.customStudent;
```

Output pane

Data Output | Explain | Messages | History

	sid bigint	name character(50)	batch batch	course character varying(10)	address character(80)	contact character(10)	email character(80)
1	201	Asim Regmi	2016	BIT	Chabahil	9807355656	asim.regmi@deerwalk.edu.np
2	205	Sagar Giri	2016	BSCCSIT	Chabahil	9807368806	sagar.giri@deerwalk.edu.np
3	206	Sameer Koirala	2016	BSCCSIT	Koteshor	9845123456	sameer.koirala@deerwalk.edu.np

Q3. Use inheritance of tables to insert values into two tables (child and parent) simultaneously.

Parent Table: (cities)

```
CREATE TABLE public.cities
(
  name text,
  population double precision,
  altitude integer
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.cities
  OWNER TO postgres;
```

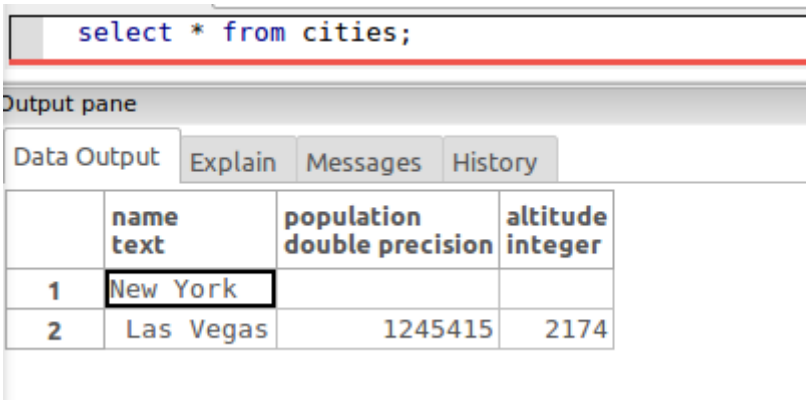
Child Table: (capitals)

```
CREATE TABLE public.capitals
(
-- Inherited from table cities:  name text,
-- Inherited from table cities:  population double precision,
-- Inherited from table cities:  altitude integer,
  state character(2)
)
INHERITS (public.cities)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.capitals
  OWNER TO postgres;
```

Inserting values into Child table:

```
INSERT INTO capitals (name, population, altitude, state)
VALUES
('New York', NULL, NULL, 'NY'),
(' Las Vegas',1245415,2174,'CC');
```

Displaying results from parent table:



Displaying results from child table:

```
select * from capitals;
```

Output pane

Data Output | Explain | Messages | History

	name text	population double precision	altitude integer	state character(2)
1	New York			NY
2	Las Vegas	1245415	2174	CC

You can also write the table name with a trailing * to explicitly specify that descendant tables are included:

```
SELECT name, altitude
    FROM cities*
    WHERE altitude > 500;
```

Output pane

Data Output | Explain | Messages | History

	name text	altitude integer
1	Las Vegas	2174

Q4. Display the stored data with Object Identifier.

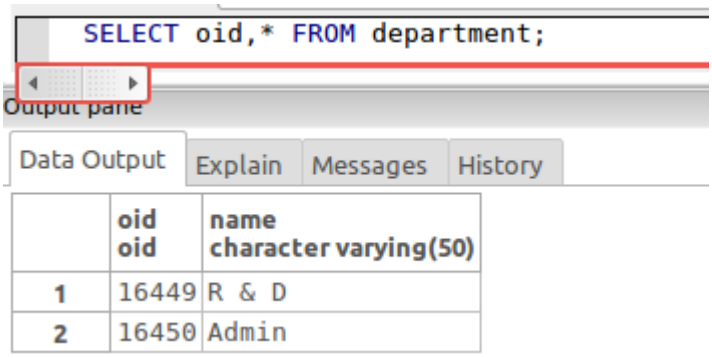
For Object identifier we need to specify to create table with OID.

```
CREATE TABLE department
(
    name varchar(50)
) WITH OIDS;
```

Insert sample values:

```
INSERT INTO department VALUES
('R & D'), ('Admin');
```

We can view data with associated OID as:



	oid oid	name character varying(50)
1	16449	R & D
2	16450	Admin