PROJECT REPORT

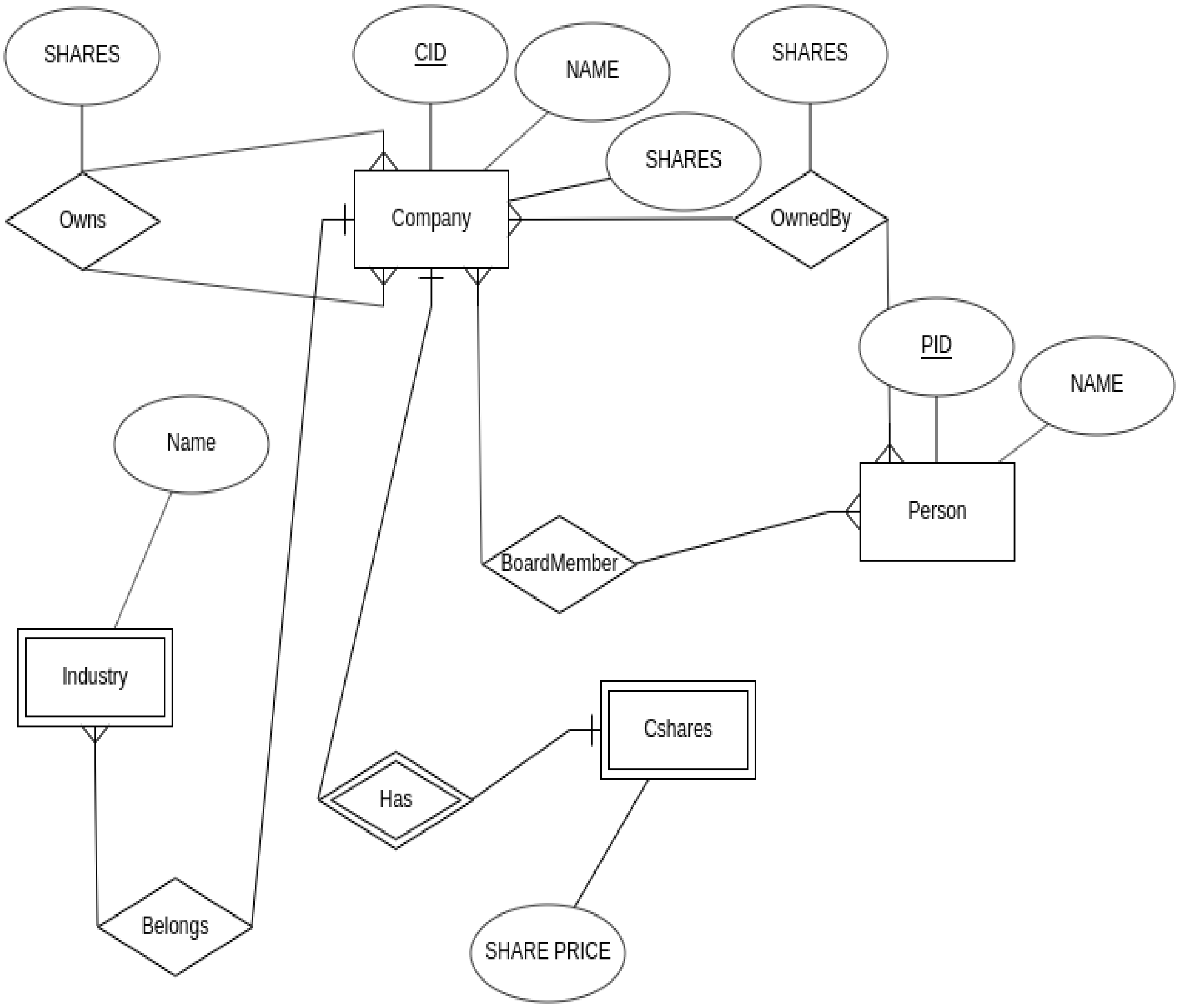
I pledge my honor that all parts of this project were done by me alone and without collaboration with anybody else.

SBUID : 112551443

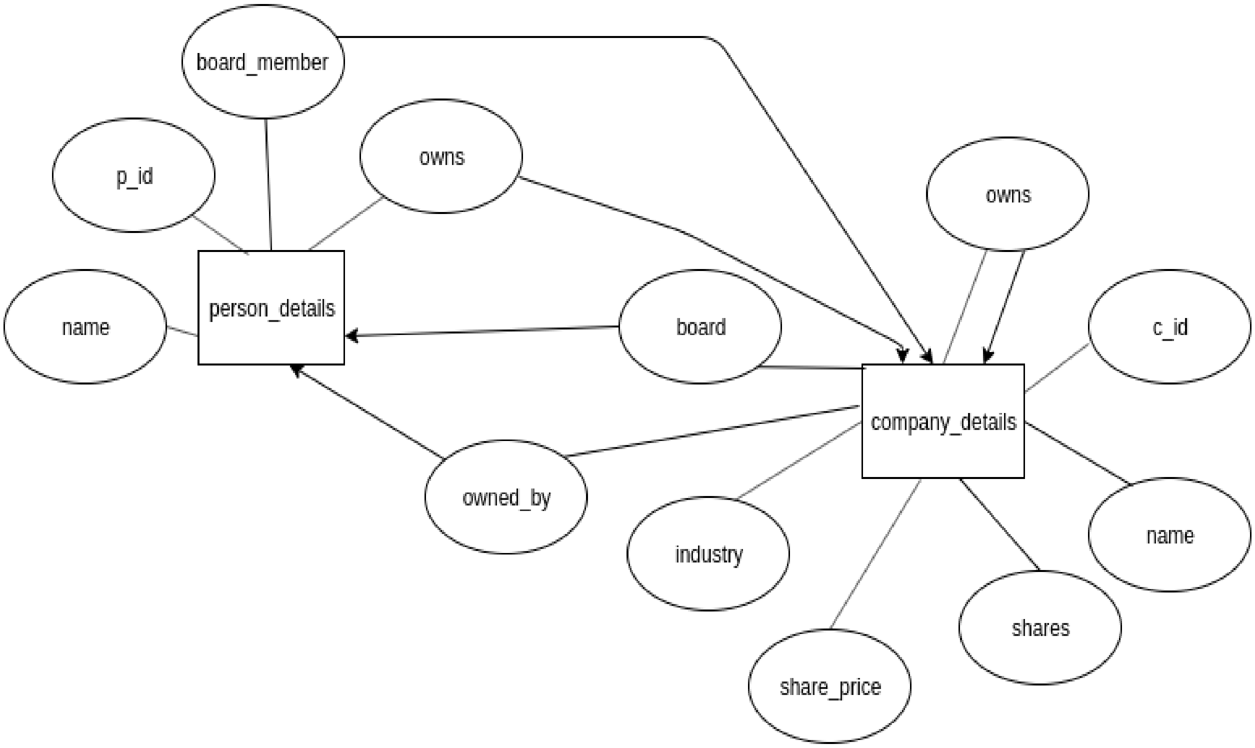
Name: Kushagra Pareek

Entity Relationship Diagram :

Relational Diagram :



Object Relational Diagram:



DESIGN DECISIONS

My database uses only two tables(objects) company_details and person_details ,

Structure of company_details

```
CREATE TABLE public.company_details
(
  c_id character varying(20) NOT NULL,
  name text NOT NULL,
  shares bigint CHECK(shares > 0),
  share_price bigint,
  owned_by dict[],
  board text[],
  owns dict[],
  industry text[],
CONSTRAINT c_id_pk PRIMARY KEY (c_id)
)
```

Structure of person_details

```
CREATE TABLE public.person_details
(
  p_id character varying(20) NOT NULL,
  name text NOT NULL,
  owns dict[],
  board_member text[],
  CONSTRAINT p_id_pk PRIMARY KEY (p_id)
)
```

Note : dict is a created type of the form

```
CREATE TYPE public.dict AS
   (id character varying(20),
    amount bigint);
```

A company has references to itself captured by owns[] of the company_details table, As postgres is limited in its features, real objects cannot be stored in the owns array, rather I store (c_id, sharest) AS dict in the owns array of each company that another company owns, I decided not to use OID feature of postgres as it is a deprecated feature and supports only four bytes. Similarly the board field of the company_details table captures objects of person_details(person type), but similar to above explanation, due to the limitations of postgres, I just capture identity of each person(p_id) from the person_details table.

The owns field of person details table captures the objects of the company_details stores as (c_id, shares) also the board_member field captures that a person is a board member of different companies.

The board field of company_details and board_member field of person_details are inverse relationships, similarly owned_by field of company_details and owns field of person_details are inverse relationships.

p_id is primary key of person_details table and c_id is primary key of company_details table.

Execution process

Apart from the database design, I created a frontend so that a user can execute a query on the database using the browser, to create that I used jsp and servlets, the servlet Query.java in my project provides the functionality to connect to database and execute queries on it, also the result from the database are sent to the server using this servlet, the Index.jsp file of the project displays the initial page when the page is first seen on the browser, whenever a query is clicked on this page, it passes on to the servlet and an appropriate query is executed and the results are passed on to the server for display.

Postgresql script

```
CREATE DATABASE woco1
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       LC_COLLATE = 'en_US.UTF-8'
       LC_CTYPE = 'en_US.UTF-8'
       CONNECTION LIMIT = -1;
--connect to database
\c woco1

--create a type dictionary

CREATE TYPE public.dict AS
   (id character varying(20),
    amount bigint);
ALTER TYPE public.dict
  OWNER TO postgres;


CREATE TABLE public.company_details
(
  c_id character varying(20) NOT NULL,
  name text NOT NULL,
  shares bigint CHECK(shares > 0),
  share_price bigint,
  owned_by dict[],
  board text[],
  owns dict[],
  industry text[],
  CONSTRAINT c_id_pk PRIMARY KEY (c_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.company_details
  OWNER TO postgres;


CREATE TABLE public.person_details
(
  p_id character varying(20) NOT NULL,
  name text NOT NULL,
  owns dict[],
  board_member text[],
  CONSTRAINT p_id_pk PRIMARY KEY (p_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.person_details
  OWNER TO postgres;


--Insert in company_details
INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c1', 'QUE', 150000, 30,
'{"(p3,20000)","(p5,50000)","(p7,30000)"}', '{p3,p1,p4}',
'{"(c2,10000)","(c4,20000)","(c8,30000)"}', '{Software,Accounting}');

INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c2', 'RHC', 250000, 20,
'{"(p3,20000)","(p4,30000)","(p5,70000)","(p6,40000)","(p7,-9000)","(p8,60000)"}',
'{p2,p1,p5}', NULL, '{Accounting}');

INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c4', 'Elgog', 1000000, 400, '{"(p2,20000)","(p7,30000)"}',
'{p5,p6,p7}', '{"(c6,5000)"}', '{Software,Search}');

INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c5', 'Tfos', 10000000, 300,
'{"(p1,30000)","(p3,800000)","(p4,40000)","(p7,300000)"}', '{p2,p4,p5}',
'{"(c6,30000)","(c7,50000)","(c1,200000)"}', '{Software,Hardware}');

INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c6', 'Ohay', 180000, 50, '{"(p5,50000)","(p8,-40000)"}',
'{p2,p4,p8}', NULL, '{Search}');
```

```
INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c7', 'Gnow', 150000, 300, '{"(p2,40000)","(p7,80000)"}',
'{p2,p3,p4}', NULL, '{Search}');
```

```
INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c8', 'Elpa', 9000000, 300,
'{"(p1,100000)","(p5,90000)","(p8,30000)"}', '{p2,p3,p8}', '{"(c4,30000)","(c5,20000)"}',
'{Software,Hardware}');
```

```
INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c9', 'Ydex', 5000000, 100, '{"(p6,-40000)","(p8,-80000)"}',
'{p6,p3,p8}', NULL, '{Software,Search}');
```

```
INSERT INTO public.company_details (c_id, name, shares, share_price, owned_by, board,
owns, industry) VALUES ('c3', 'Alf', 10000000, 700, '{"(p4,500000)","(p6,500000)"}',
'{p6,p7,p1}', '{"(c9,-100000)","(c4,400000)","(c8,100000)"}', '{Software,Automotive}');
```

Insert in person_details

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p1', 'Bill
Doe', '{"(c5,30000)","(c8,100000)"}', '{c1,c2,c3}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p2', 'Bill
Seth', '{"(c7,40000)","(c4,20000)"}', '{c2,c5,c6,c7,c8}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p3', 'John
Smyth', '{"(c1,20000)","(c2,20000)","(c5,800000)"}', '{c1,c7,c8,c9}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p4', 'Anne
Smyle', '{"(c2,30000)","(c5,40000)","(c3,500000)"}', '{c1,c5,c6}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p5', 'Steve
Lamp', '{"(c8,90000)","(c1,50000)","(c6,50000)","(c2,70000)"}', '{c2,c4,c5}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p6', 'May
Serge', '{"(c8,-10000)","(c9,-40000)","(c3,500000)","(c2,40000)"}', '{c3,c4,c9}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p7', 'Bill
public', '{"(c7,80000)","(c4,30000)","(c1,30000)","(c5,300000)","(c2,-9000)"}', '{c3,c4}');
```

```
INSERT INTO public.person_details (p_id, name, owns, board_member) VALUES ('p8', 'Muck
Lain', '{"(c2,60000)","(c6,-40000)","(c9,-80000)","(c8,30000)"}', '{c6,c8,c9}');
```

Queries :

```
/**Query-1**/

CREATE OR REPLACE FUNCTION get_company_name(id varchar(20))
RETURNS text as $$
BEGIN
 RETURN (SELECT name from company_details where id = c_id);
END; $$
LANGUAGE plpgsql;

SELECT get_company_name(O.c_id)
FROM person_details P, unnest(P.owns) AS O(c_id, amount), unnest(P.board_member) AS
B(id)
where O.c_id = B.id and O.amount > 0


/**Query-2**/

CREATE OR REPLACE FUNCTION get_share_price(s varchar(20))
RETURNS bigint AS $$
BEGIN
 RETURN (SELECT share_price from company_details where c_id = s);
END; $$
LANGUAGE plpgsql;

SELECT P.name AS name , sum(O.amount * get_share_price(O.c_id)) AS networth FROM
person_details P, unnest(P.owns) AS O(c_id, amount)
WHERE O.amount > 0
GROUP BY P.p_id

/**Query-3**/

CREATE OR REPLACE FUNCTION get_person_name(id varchar(20))
RETURNS text as $$
BEGIN
 RETURN (SELECT name from person_details where p_id = id);
END; $$
LANGUAGE plpgsql;

SELECT top_members.company , top_members.TopBoardMember
FROM
(SELECT C.name AS company, get_person_name(O.p_id) AS TopBoardMember,
max(O.amount)
FROM company_details C, unnest(C.owned_by) AS O(p_id, amount), unnest(C.board) AS
B(id)
WHERE O.p_id = B.id and O.amount > 0
GROUP BY C.c_id, O.p_id) top_members
```

```
/**Query-4**/

SELECT c1.name AS cname1 , c2.name AS cname2
FROM company_details c1, company_details c2
WHERE c1.c_id <> c2.c_id
AND
  EXISTS( SELECT 1 FROM
   unnest(c1.industry) ind1, unnest(c2.industry) ind2
   WHERE ind1 = ind2) AND
   NOT EXISTS ( (

                  SELECT p.p_id , p.cmp FROM
                   (SELECT p_id , O.cmp, O.shrs FROM
                       person_details, unnest(owns) AS O(cmp, shrs) ) p
                  WHERE p.p_id = ANY(c2.board) and p.shrs > 0 )

                  EXCEPT (
                          SELECT p2.p_id , p2.cmp FROM
                         ( SELECT p_id, O.cmp, O.shrs
                      FROM person_details, unnest(owns) AS O(cmp, shrs) ) p2
                          WHERE p2.p_id = ANY(c2.board) and p2.shrs > 0 AND
                          p2.shrs  <= ANY( SELECT p1.shrs FROM
                                             ( SELECT p_id, O.cmp, O.shrs
                         FROM person_details, unnest(owns) AS O(cmp , shrs) ) p1
                WHERE p1.p_id = ANY(c1.board) AND   p2.cmp = p1.cmp  AND p1.shrs > 0 ) ) ) ;


/**Query-5**/

CREATE OR REPLACE FUNCTION get_shares(id varchar(20))
RETURNS bigint as $$
BEGIN
 RETURN (SELECT shares from company_details where id = c_id);
END; $$
LANGUAGE plpgsql;

CREATE OR REPLACE VIEW directOwns AS
(SELECT C.c_id AS id, O.c_id AS own_id, ROUND((O.amount * 1.0/get_shares(O.c_id)),25) AS
fraction
From company_details C, unnest(C.owns) AS O(c_id, amount)
WHERE O.amount > 0);

CREATE OR REPLACE RECURSIVE VIEW  indirectOwns(id, own_id, fraction) AS(
      SELECT * FROM directOwns
      UNION
      SELECT D.id , I.own_id, ROUND(D.fraction * I.fraction , 25)
      FROM directOwns D, indirectOwns I
```

```
        WHERE D.own_id = I.id AND D.id <> I.own_id);


CREATE OR REPLACE VIEW directpOwns AS
(SELECT P.p_id AS id, O.c_id AS own_id, ROUND((O.amount * 1.0/get_shares(O.c_id)),25) AS
fraction
From person_details P, unnest(P.owns) AS O(c_id, amount)
WHERE O.amount > 0);



CREATE OR REPLACE RECURSIVE VIEW  indirectpOwns(id, own_id, fraction) AS(
        SELECT * FROM directpOwns
        UNION
        SELECT D.id , I.own_id, ROUND(D.fraction * I.fraction , 25)
        FROM indirectpOwns D, indirectOwns I   WHERE D.own_id = I.id );
CREATE OR REPLACE VIEW tb AS (
select id, own_id , sum(fraction) AS frac from indirectpOwns
 GROUP BY id,own_id );

select get_person_name(id) AS NAME, get_company_name(own_id) AS CNAME, (frac*100)
AS PERCENT from tb where frac > 0.1;
```

USER GUIDE

Deploying on web server (Linux based OS)

Note: you should have a web server (Tomcat).

1. Download and Install postgresql.
2. Download and Install JAVA version 8 or above.
3. Create the woco1 database using the file provided woco1.sql in the zip folder.
4. Use command sudo -u postgres psql postgres -f   “/filelocation/woco1.sql”
5. The above step will create the database and Insert data into the database.
6. Now from the zip file copy .war file and place it on */apache-tomcat-<version>/webapps/* folder.
7. To start the tomcat server, go to */apache-tomcat-9.0.0.M10/bin and do ./catalina.sh start*
8. You will be able to see the index page on the public DNS.