

Project #2

CSE532 – Project 2

File name: Project 2 Report

Author(s): Sravya Beesabathuni (111327265)

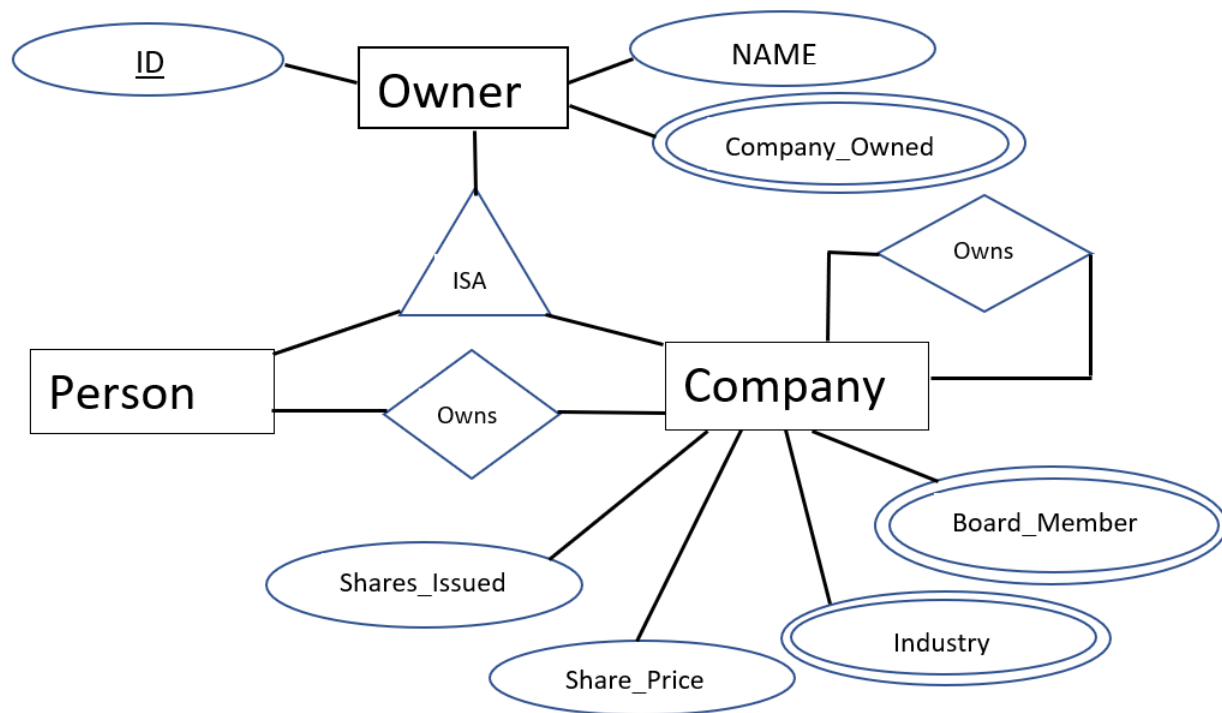
Pooja R Dalaya (111323959)

Brief description: Project Report which contains ER diagram, DDL and DML commands, User Guide and the queries

/*We pledge our honor that all parts of this project were done by us alone and without collaboration with anybody else.*/*

1 Entity Relationship Diagram:

The Entity Relationship Diagram is given below:



2 Description of the Database Schema:

The database schema has been designed in an object-oriented way. The schema consists of three tables Owner, Company and Person. As Company and Person have common features that Owner has, we have established a hierarchical structure using the Inheritance concept supported by PostGreSQL.

The description of all the entities is given below:- (On running \d+)

S - Stats Target

D - Description

Table "public.owner"

Column	Type	Collation	Nullable	Default	Storage	S	D
id	integer		not null		plain		
name	text		not null		extended		
company_owned_list	company_owned[]				extended		

Indexes: "owner_pkey" PRIMARY KEY, btree (id)

Child tables: company, person

Table "public.person"

Column	Type	Collation	Nullable	Default	Storage	S	D
id	integer		not null		plain		
name	text		not null		extended		
company_owned_list	company_owned[]				extended		

Inherits Owner

Table "public.company"

Column	Type	Collation	Nullable	Default	Storage	S	D
id	integer		not null		plain		
name	text		not null		extended		
company_owned_list	company_owned[]				extended		
shares_issued	numeric		not null		main		
share_price	real		not null		plain		
industry	text[]				extended		
board_members	integer[]				extended		

Inherits Owner

We can clearly see from the Schema that Person and Company are child tables where as Owner is the parent table. While populating values in the Child

table, the values were automatically inserted into the parent table. We also made use of a UDT `company_owned`.

Composite type "public.company_owned"

Column	Type	Collation	Nullable	Default	Storage	Description
id	integer				plain	
shares_owned	numeric				main	

3 Integrity Constraints:

We have the following integrity constraints on the tables/types explained above in the schema.

3.1 OWNER Table:

- **ID** has **Primary Key** and **NOT NULL** constraints.
- **NAME** has **NOT NULL** constraint.

3.2 COMPANY Table Inherits OWNER:

- **SHARES_ISSUED** has **NOT NULL** constraint.
- **SHARE_PRICE** has **NOT NULL** constraint.
- **INDUSTRY** has **NOT NULL** constraint.
- **BOARD_MEMBERS** has **NOT NULL** constraint.

3.3 PERSON Inherits COMPANY

4 DDL Commands:

4.1 Types :

4.1.1 COMPANY_OWNED

```
DROP TYPE IF EXISTS COMPANY_OWNED;
```

```
CREATE TYPE COMPANY_OWNED AS (  
    Id int NOT NULL,  
    Shares_Owned numeric  
);
```

4.2 Tables:

4.2.1 OWNER

```
DROP TABLE IF EXISTS OWNER;
```

```
CREATE TABLE OWNER(  
    Id int PRIMARY KEY NOT NULL,  
    Name text NOT NULL,  
    company_owned_list company_owned[]    );
```

4.2.2 PERSON

```
DROP TABLE IF EXISTS PERSON;
```

```
CREATE TABLE PERSON() INHERITS (OWNER);
```

4.2.3 COMPANY

```
DROP TABLE IF EXISTS COMPANY;
```

```
CREATE TABLE COMPANY(  
    shares_issued numeric NOT NULL,  
    share_price numeric NOT NULL,  
    board_members integer[],  
    industry text[]  
) INHERITS (OWNER);
```

5 QUERIES:

5.1 Query 1

```
SELECT DISTINCT(C.NAME) AS CNAME  
FROM PERSON P, COMPANY C, UNNEST(P.COMPANY_OWNED_LIST)  
AS PERSONOWNS, UNNEST(C.BOARD_MEMBERS) AS BM  
WHERE BM=P.ID AND C.ID=PERSONOWNS.ID AND PERSONOWNS.SHARES_OWNED  
> 0  
ORDER BY C.NAME;
```

5.2 Query 2

```
SELECT P.NAME, SUM(COMPLIST.SHARES_OWNED*C.SHARE_PRICE)  
AS NETWORTH  
FROM PERSON P, COMPANY C, UNNEST(P.COMPANY_OWNED_LIST)  
AS COMPLIST  
WHERE COMPLIST.ID = C.ID AND COMPLIST.SHARES_OWNED > 0
```

```
GROUP BY P.NAME
ORDER BY P.NAME;
```

5.3 Query 3

```
SELECT C.NAME AS COMPANYNAME, P.NAME AS PERSONNAME
FROM COMPANY C, PERSON P, UNNEST(C.BOARD_MEMBERS) AS BM,
UNNEST(P.COMPANY_OWNED_LIST) AS COMPLIST
WHERE P.ID = BM AND COMPLIST.ID = C.ID AND COMPLIST.SHARES_OWNED
IN (
    SELECT MAX(COMPLIST.SHARES_OWNED)
    FROM PERSON P, COMPANY C, UNNEST(P.COMPANY_OWNED_LIST)
    AS COMPLIST , UNNEST(C.BOARD_MEMBERS) AS BM
    WHERE BM=P.ID AND COMPLIST.ID = C.ID AND COMPLIST.SHARES_OWNED>0
    GROUP BY C.ID
)
ORDER BY C.NAME;
```

5.4 Query 4:

```
SELECT C1.NAME AS C1NAME,C2.NAME AS C2NAME
FROM COMPANY C1, COMPANY C2
WHERE C1.INDUSTRY && C2.INDUSTRY AND C1.ID<>C2.ID AND
NOT EXISTS(
    SELECT C.ID
    FROM COMPANY C, PERSON P, UNNEST(C2.BOARD_MEMBERS)
    AS BM2, UNNEST(P.COMPANY_OWNED_LIST) AS COLIST2
    WHERE COLIST2.ID=C.ID AND P.ID=BM2
EXCEPT
    SELECT CO.ID
    FROM COMPANY CO,UNNEST(C1.BOARD_MEMBERS) AS B1,
    UNNEST(C2.BOARD_MEMBERS) AS B2, PERSON P1, PERSON P2,
    UNNEST(P1.COMPANY_OWNED_LIST) AS CO1, UNNEST(P2.COMPANY_OWNED_LIST)
    AS CO2
    WHERE P1.ID=B1 AND P2.ID=B2 AND CO.ID=CO1.ID AND CO.ID=CO2.ID
    GROUP BY CO.ID,CO1.SHARES_OWNED
    HAVING CO1.SHARES_OWNED≥MAX(CO2.SHARES_OWNED)
);
```

5.5 Query 5:

```
WITH RECURSIVE PERSON_INDIRECTLY_OWNS AS (
    SELECT P.ID AS PERSONID, COMPLIST.ID AS COMPANYID,
    ROUND((COMPLIST.SHARES_OWNED/C.SHARES_ISSUED),6) AS VAL
    FROM COMPANY C, PERSON P, UNNEST(P.COMPANY_OWNED_LIST)
    AS COMPLIST
```

```

WHERE COMPLIST.ID = C.ID AND COMPLIST.SHARES_OWNED
> 0
UNION
SELECT PO.PERSONID AS PERSONID, CO.COMPANY2 AS
COMPANYID, ROUND((PO.VAL*CO.VAL),6) AS VAL
FROM PERSON_INDIRECTLY_OWNS PO, COMPANY_INDIRECTLY_OWNS
CO
WHERE PO.COMPANYID = CO.COMPANY1 AND CO.VAL > 0
),
COMPANY_INDIRECTLY_OWNS AS (
SELECT C.ID AS COMPANY1, CLIST.ID AS COMPANY2,
ROUND((CLIST.SHARES_OWNED/C2.SHARES_ISSUED), 6) AS VAL
FROM COMPANY C, COMPANY C2, UNNEST(C.COMPANY_OWNED_LIST) CLIST
WHERE C2.ID = CLIST.ID AND CLIST.SHARES_OWNED > 0
UNION
SELECT C.ID AS COMPANY1, CO.COMPANY2 AS COMPANY2,
ROUND((CO.VAL *(CLIST.SHARES_OWNED/C2.SHARES_ISSUED)),6)
AS VAL
FROM COMPANY C, COMPANY_INDIRECTLY_OWNS CO,
COMPANY C2, UNNEST(C.COMPANY_OWNED_LIST) CLIST
WHERE CO.COMPANY1 = CLIST.ID AND CLIST.ID = C2.ID AND
CLIST.SHARES_OWNED > 0 AND CO.VAL > 0
)
SELECT PER.NAME, C.NAME, SUM(P.VAL*100) AS PERCENTAGE
FROM PERSON_INDIRECTLY_OWNS P, PERSON PER, COMPANY C
WHERE PER.ID = P.PERSONID AND C.ID = P.COMPANYID
GROUP BY PER.NAME, C.NAME
HAVING SUM(P.VAL*100) > 10
ORDER BY PER.NAME;

```

6 User Guide:

For the functioning of this Project the following components have been installed:

1. PostgreSQL Database: We have installed PostgreSQL 10.1 from the official EDB site. User is default "postgres" and the service runs at port 5432.
2. Eclipse Oxygen IDE integrated with JDK 8(Only Oxygen has support for Tomcat 9)
3. Apache Tomcat v 9.0 with the service running on port 9001
4. Downloaded the Postgres JDBC driver (postgresql-42.1.4)

The post installation steps include:

1. Setting the appropriate class paths for CATALINA_HOME

2. Copying the postgres JDBC jar file in the lib folder of Apache.
3. Importing postgres JDBC jar file in the Web Project in Java(As an external jar file).
4. Compile the Java code and place the .class file in `\%APACHE_HOME%\apache-tomcat-9.0.1\webapps\ROOT\WEB-INF\classes`
5. Provide the proper servlet mapping in web.xml

We have made use of a JSP file named WOCO.jsp which is responsible for the building of the basic UI which triggers the appropriate query via the Servlet by using the parameters. The servlet then executes the appropriate query and builds the response JSP with a BACK link.

Thus the project can be run via Eclipse, i.e. Right-click WOCO.jsp file and run on server. It automatically launches the Main webpage from where we can launch various links to execute all the five queries.

7 Teaming:

We have collaborated right from the beginning to design the Entity-Relationship diagram, Database Design and also the Report.

Individual work done:

- Pooja R Dalaya: I have worked on the following components of the Project:
 1. Query 1 and 4.
 2. Implemented the logic to link the Servlet to the PostgreSQL Database.
- Sravva Beesabathuni:
 1. Worked on Query 2,3 and 5.
 2. Coded the JSP file to display the query result when clicked on the Query . Also worked on JSP to Servlet connection.