**Fall 2019 CSE 532: Project 2**

**Implementing WOCO Database using Object Oriented Extensions of SQL**

**Meghana Vemulapalli Harshini Kanaparthi**

**112670836 112687951**

[**Meghana.vemulapalli@stonybrook.edu**](mailto:Meghana.vemulapalli@stonybrook.edu)[**harshini.kanaparthi@stonybrook.edu**](mailto:harshini.kanaparthi@stonybrook.edu)

**Github link:** https://github.com/kifer-SBCS/cse532-project2-db\_meghana\_harshini.git

I pledge my honor that all parts of this project were done by us and without collaboration with anybody else. I did not use any external code that is directly related to this assignment.

**Table of Contents**

Topics Pages

Project Description 3

ER Diagram 3

Database Design 3

Integrity Constraints 4

Indexing 4

SQL Create/ Insert Queries 4

Queries for Questions 1 to 5 6

User Guide 7

Division of Work 7

**PROJECT DESCRIPTION:**

The WOCO database contains information about companies, owners (based on the number of shares owned), and board members (of those companies). It is assumed that you are familiar with these concepts but here is a brief reminder.

A company has market value (number of issued stock shares times share price). Companies belong to one or more industries. Companies have owners, where an owner is someone who holds some company shares. An owner can be a person or a company. A company can have a board whose members are people charged with deciding on the overall direction of the company.

**ER DIAGRAM:**



**DATABASE SCHEMA DESIGN:**

1. Entity – of type EntityType which has attributes id (primary key) and name – name of entity.
2. Company – inherits from Entity which has attributes id (primary key) , industry (array of type TEXT) and ownership (array of type ownershipType which has attributes cid and sharesowned).
3. Shares – has attributes cid (primary key and also a foreign key referring to id in company), shareprice and totalsharevalue of a company.
4. Person - inherits from Entity which has attributes id (primary key) and ownership (array of type ownershipType which has attributes cid and sharesowned).
5. Board – has attributes pid (foreign key referring to id in person) and cid (foreign key referring to id in company). As each cid can have multiple persons in board, primary key in Board – (cid, pid).

All the tables are normalized, and schema is in 4NF. In the above design, we have many kinds of relationships:

1. A person can own shares in multiple companies, so relationship to person and ownership is 1 : M.
2. A company can own shares in multiple companies, so relationship to company and ownership is 1 : M.
3. One company has a board, so relationship from company to board is 1 : 1.
4. There can exist multiple persons in board and a person can belong to multiple boards of different companies. So relationship between person and board is M:N.
5. A company can have single sharevalue and totalshares so relationship between company and shares is 1 : 1.

**INTEGRITY CONSTRAINTS:**

Below are the various integrity constraints used in designing database schema:

Primary Key: id in Entity, id in Company, id in Person, cid in Shares, (cid, pid) in Board.

Foreign Key: cid in Shares referring to id in Company, cid and pid referring to id of Company and id of Person.

Not NULL: sharevalue in shares, totalshareprice in shares, name in Entity, ownedshares in ownership.

Check constraints: sharevalue > 0 in shares, totalshareprice > 0 in shares.

**SQL CREATE QUERIES:**

create type entityType as (id text, name text);

create table entity of entityType (id PRIMARY KEY, name NOT NULL);

create type ownershipType as (cid text, ownedshares float);

create table person (PRIMARY KEY (id), ownership ownershipType[]) inherits (entity);

create table company(primary key(id), industry text[], ownership ownershipType[]) inherits(entity);

create table shares(cid text references company(id), totalsharevalue FLOAT, shareprice FLOAT, primary key(cid));

create table board(cid text references company(id), pid text references person(id), primary key(cid,pid));

**INDEXING:**

As we’re performing most of the operations on ownedshares in ownership like getting the maximum value or validating ownedshares to be positive, so we can arrange in Decreasing order for better performance.

CREATE INDEX person\_index ON person (ownership DESC NULLS LAST);

CREATE INDEX company\_index ON company (ownership DESC NULLS LAST);

**SQL INSERT QUERIES:**

**Inserting queries into person table**

INSERT INTO PERSON VALUES ('P1','Bill Doe', ARRAY[('C5',30000),('C8',100000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P1','Bill Doe', ARRAY[('C5',30000),('C8',100000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P2','Bill Seth', ARRAY[('C7',40000),('C4',20000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P3','John Smyth', ARRAY[('C1',20000),('C2',20000),('C5',800000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P4','Anne Smyle', ARRAY[('C2',30000),('C5',40000),('C3',500000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P5','Steve Lamp',ARRAY[('C8',90000),('C1',50000),('C6',50000),('C2',70000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P6','May Serge', ARRAY[('C8',-10000),('C9',-40000),('C3',500000),('C2',40000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P7','Bill Public', ARRAY[('C7',80000),('C4',30000),('C1',30000),('C5',300000),('C2',-9000)]::ownershipType[]);

INSERT INTO PERSON VALUES ('P8','Muck Lain', ARRAY[('C2',60000),('C6',-40000),('C9',-80000),('C8',30000)]::ownershipType[]);

**Inserting queries into company table**

INSERT INTO company VALUES ('C1','QUE', ARRAY[('Software'),('Accounting')]::TEXT[],ARRAY[('C2',10000),('C4',20000),('C8',30000)]::ownershipType[]);

INSERT INTO company VALUES ('C2','RHC', ARRAY[('Accounting')]::TEXT[],ARRAY[]::ownershipType[]);

INSERT INTO company VALUES ('C3','Alf', ARRAY[('Software'),('Automotive')]::TEXT[],ARRAY[('C9',- 100000),('C4',400000),('C8',100000)]::ownershipType[]);

INSERT INTO company VALUES ('C4','Elgog', ARRAY[('Software'),('Search')]::TEXT[],ARRAY[('C6',5000)]::ownershipType[]);

INSERT INTO company VALUES ('C5','Tfos', ARRAY[('Software'),('Hardware')]::TEXT[],ARRAY[('C1',200000),('C6',30000),('C7',50000)]::ownershipType[]);

INSERT INTO company VALUES ('C6','Ohay', ARRAY[('Search')]::TEXT[], ARRAY[]::ownershipType[]);

INSERT INTO company VALUES ('C7','Gnow', ARRAY[('Search')]::TEXT[],ARRAY[]::ownershipType[]);

INSERT INTO company VALUES ('C8','Elpa', ARRAY[('Software'),('Hardware')]::TEXT[], ARRAY[('C5',20000),('C4',30000)]::ownershipType[]);

INSERT INTO company VALUES ('C9','Ydex', ARRAY[('Software'),('Search')]::TEXT[], ARRAY[]::ownershipType[]);

**Inserting queries into shares table**

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C1',150000, 30);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C2',250000, 20);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C3',10000000,700);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C4',1000000,400);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C5',10000000,300);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C6',180000,50);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C7',150000,300);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C8',9000000,300);

INSERT INTO SHARES(CID, TOTALSHAREVALUE, SHAREPRICE) VALUES ('C9',5000000,100);

**Inserting queries into board table**

INSERT INTO BOARD(CID, PID) VALUES ('C1','P3');

INSERT INTO BOARD(CID, PID) VALUES ('C1','P1');

INSERT INTO BOARD(CID, PID) VALUES ('C1','P4');

INSERT INTO BOARD(CID, PID) VALUES ('C2','P1');

INSERT INTO BOARD(CID, PID) VALUES ('C2','P2');

INSERT INTO BOARD(CID, PID) VALUES ('C2','P5');

INSERT INTO BOARD(CID, PID) VALUES ('C3','P6');

INSERT INTO BOARD(CID, PID) VALUES ('C3','P7');

INSERT INTO BOARD(CID, PID) VALUES ('C3','P1');

INSERT INTO BOARD(CID, PID) VALUES ('C4','P6');

INSERT INTO BOARD(CID, PID) VALUES ('C4','P7');

INSERT INTO BOARD(CID, PID) VALUES ('C4','P5');

INSERT INTO BOARD(CID, PID) VALUES ('C5','P2');

INSERT INTO BOARD(CID, PID) VALUES ('C5','P5');

INSERT INTO BOARD(CID, PID) VALUES ('C5','P4');

INSERT INTO BOARD(CID, PID) VALUES ('C6','P2');

INSERT INTO BOARD(CID, PID) VALUES ('C6','P4');

INSERT INTO BOARD(CID, PID) VALUES ('C6','P8');

INSERT INTO BOARD(CID, PID) VALUES ('C7','P2');

INSERT INTO BOARD(CID, PID) VALUES ('C7','P3');

INSERT INTO BOARD(CID, PID) VALUES ('C7','P4');

INSERT INTO BOARD(CID, PID) VALUES ('C8','P2');

INSERT INTO BOARD(CID, PID) VALUES ('C8','P3');

INSERT INTO BOARD(CID, PID) VALUES ('C8','P8');

INSERT INTO BOARD(CID, PID) VALUES ('C9','P3');

INSERT INTO BOARD(CID, PID) VALUES ('C9','P6');

INSERT INTO BOARD(CID, PID) VALUES ('C9','P8');

**QUERY FOR QUESTION 1:**

SELECT comp.name

FROM company comp, board b, person per, unnest(per.ownership) as owns

WHERE per.id = b.pid and comp.id = owns.cid and comp.id = b.cid and owns.ownedshares>0

ORDER BY comp.name;

**QUERY FOR QUESTION 2:**

SELECT per.name, sum(owns.ownedshares \* s.shareprice)

FROM person per, company comp, unnest(per.ownership) as owns, shares s

WHERE owns.cid = comp.id and comp.id = s.cid and owns.ownedshares > 0

GROUP BY per.name

ORDER BY per.name;

**QUERY FOR QUESTION 3:**

SELECT comp.name, per.name   
FROM company comp, person per, board b, unnest(per.ownership) as owns   
WHERE b.pid = per.id and comp.id = b.cid and owns.cid = comp.id

and owns.ownedshares > 0 and owns.ownedshares =

(SELECT max(own.ownedshares)   
FROM person p, unnest(p.ownership) as own

WHERE b.pid = p.id and own.cid = comp.id and own.ownedshares>0)   
ORDER BY comp.name;

**QUERY FOR QUESTION 4:**  
  
SELECT comp1.name, comp2.name   
FROM company comp1, company comp2, unnest(comp1.industry) as indus1,

unnest(comp2.industry) as indus2   
WHERE [comp1.id](http://comp1.id) <> [comp2.id](http://comp2.id) and indus1=indus2 and

NOT EXISTS(SELECT \* FROM person per2, board b, unnest(per2.ownership) as owns   
      WHERE [comp2.id](http://comp2.id) = b.cid and [per2.id](http://per2.id) = b.pid and owns.ownedshares > 0 and

NOT EXISTS(SELECT \*   
    FROM person per1, board b1, unnest(per1.ownership) as own   
                          WHERE [comp1.id](http://comp1.id) = b1.cid and b1.pid = [per1.id](http://per1.id) and

own.cid = owns.cid and own.ownedshares >= owns.ownedshares))

GROUP BY [comp1.name](http://comp1.name), [comp2.name](http://comp2.name);

**QUERY FOR QUESTION 5:**

create recursive view direct\_ownership(personid, cid, percent) as

(select comp2.id, comp1.id, cast(cast(owns.ownedshares as FLOAT)/cast(s.totalsharevalue as FLOAT) as numeric(10,8)) from company comp1, company comp2, shares s, unnest(comp2.ownership) as owns where owns.cid = comp1.id and comp1.id = s.cid and owns.ownedshares>0)

UNION

(select p.id, c1.id, cast(cast(own.ownedshares as FLOAT)/cast(s.totalsharevalue as FLOAT) as numeric(10,8)) from company c1, person p, unnest(p.ownership) as own, shares s

where own.cid = c1.id and s.cid = c1.id and own.ownedshares>0);

create recursive view indirect\_ownership(entity, cid, perc) as

(select \* from direct\_ownership d)

UNION

select d.personid, i.cid, cast((d.percent\*i.perc) as numeric(10,8)) from direct\_ownership d, indirect\_ownership i where d.cid = i.entity;

with control\_per as

(select p.name as pname, c.name as cname, cast(sum(ind.perc\*100) as numeric(10,4)) as

percentage from person p, company c, indirect\_ownership ind

where p.id = ind.entity and c.id = ind.cid group by p.name, c.name)

select \* from control\_per where percentage > 10 order by pname;

**USER GUIDE:**

1. Setting up WOCO Database:

Database name : postgres

Database UserName: postgres

Database Password: postgres

1. Copy the DatabaseConnection.java, PostgresSqlServlet.java, insert.jsp, and web.xml
2. Create a java web project.
3. Create a package named Postgres under the src folder and copy DatabaseConnection.java, PostgresSqlServlet.java to the postgres package folder.
4. Copy the insert.jsp to the WebContent folder.
5. Copy web.xml to the WEB-INF sub folder under the Web Content folder.
6. Run the insert.jsp file on the tomcat server on the port 8080.
7. Proceed clicking the corresponding query buttons and ok button to go to the main page.

**DIVISION OF WORK:**

|  |  |
| --- | --- |
| **Meghana Vemulapalli** | **Harshini Kanaparthi** |
| Database schema design and Implementation | Database schema design and Implementation |
| Implementation of Postgres queries 3, 5 | Implementation of Postgres queries 1,2,4 |
| Documentation part for Database Design, Integrity Constraints, Indexing and ER diagram | Documentation of DB schema diagram, Queries, Table creation, Insertions. |
| Creating the Workflow between the queries in the servlet, JSP and JDBC. CSS Formatting and display. | Connection establishment between Postgres JDBC driver and the servlet queries. CSS Formatting |
| Creation of JSP Index file | Creation of the Java servlet file |