University of British Columbia, Department of Computer Science

CPSC 304: Project Formal Proposal

Platform:

Instead of using the provided CS UGrad Oracle installation and PHP for our application, we will be using Windows Presentation Foundation (WPF) and C# due to prior knowledge of these tools.

Functionality:

Our application will have two different types of users: Managers and Regular users, each with slightly different interfaces to the data. Regular users will only be able to query project boards on which they have membership, including all corresponding lists and tasks. Regular user queries will allow them to search for particular lists, tasks and project boards - for example, if a user would like to see which tasks they have coming up in the next week, they are able to query for such information. Regular users are able to create both lists and tasks (but are not able to delete either), move tasks between lists and update any existing tasks. They will be able to update information on tasks (task name, due date, description), change the priority and type of different lists and update the project board itself (change its name). Regular users are not able to delete project boards, lists or tasks, but they are able to create new lists and tasks.

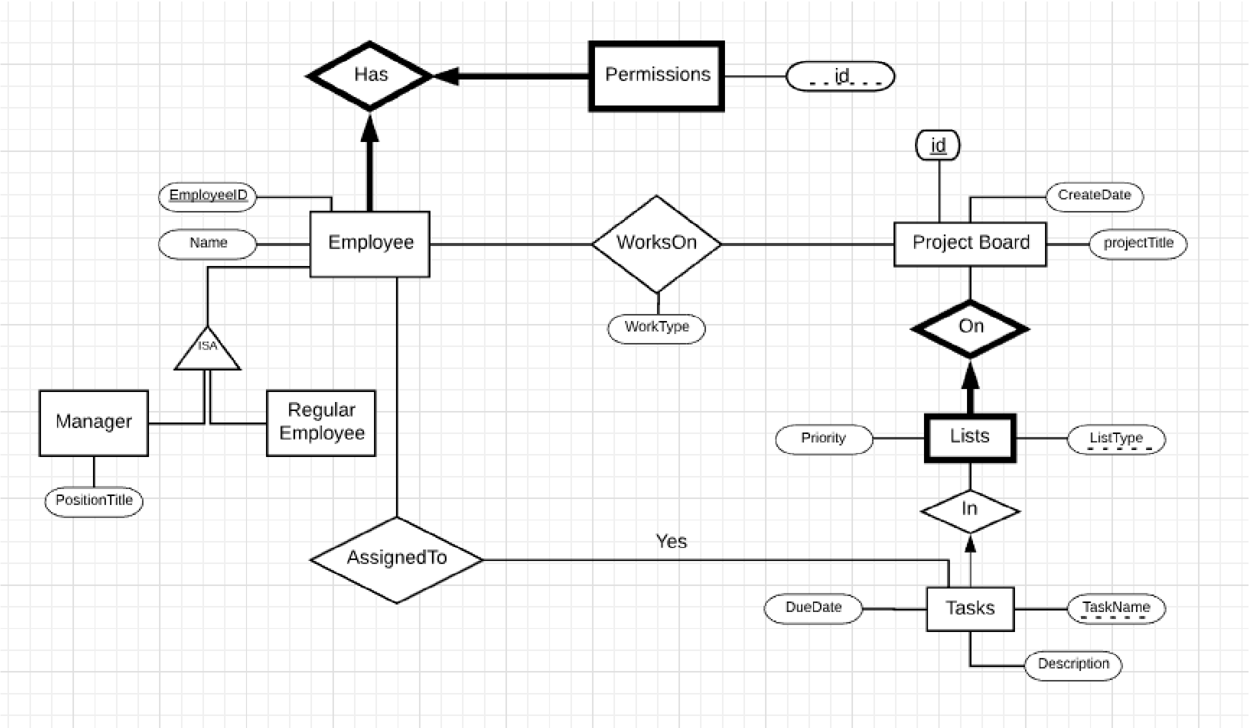
Managers will have the additional ability to create new project boards, add/create new members to each project board, query all project boards regardless of membership and delete lists and tasks. All queries available to regular users are also available to managers. Only managers will be able to update user information - create/delete users, update user information, etc.

Managers will also be able to make advanced queries across all project boards. An example is searching for which tasks are due on a particular day across all lists on all project boards. Employees will have similar yet more limited functionality, only able to query across projects boards on which they have membership.

Data:

Our database will include:

- Login information for all of our users (both Managers and Regular users)
- All project boards which have been created and all project board attributes
- All Lists across every existing project board, along with all List attributes
- All Tasks under each existing List, along with all Task attributes
- All members on on all projects



Dependencies:

taskID -> EmployeeID
taskID -> listTitle
listTitle -> pID
taskID -> pID (Armstrong Axiom of transivity)

Queries:

```
CREATE TABLE TasksIn (
	int tID PRIMARY KEY,
	date DueDate,
	text Description,
	Int listID,
	Int pID,
	FOREIGN KEY (pID, listID) REFERENCES List(pID, ListID)
)


CREATE TABLE AssignedTo(
	Int eID,
	Int taskID,
	PRIMARY KEY (eID, taskID),
	FOREIGN KEY eID REFERENCES Employee (eID),
	FOREIGN KEY taskID REFERENCES Tasks (taskID),
)
```

2: List of all members of a project board

```
    SELECT      R.name
    FROM        RegularW R, WorksOn W
    WHERE       R.eID = W.eID AND pID = _____
```

3: Coworker wants to see all the project-boards they're on

```
    SELECT      p.title
    FROM        ProjectBoard P, WorksOn W, Employee E
    WHERE       E.eID = W.eID AND W.pID = p.pID
```

4: Employee wants to see all tasks they have due before July 1st

```
    SELECT      T.TaskTitle,
    FROM        Task T, List L, ProjectBoard P, Employee E, WorksOn W
    WHERE       E.eID = W.eID AND W.pID = P.pID AND L.pID = P.pID AND T.lID = L.lID
AND T.dueDate < ____
```

5: Employee wants to add a task to a project board

```
    INSERT INTO Task(pID, listID, taskID, taskTitle, dueDate)
    VALUES (___,___,___,___,___)
```

6: Manager wants to see all tasks over all project boards due within a certain time frame

```
    SELECT      T.taskTitle
    FROM Task T
    WHERE       T.dueDate > _____ AND T.dueDate <= ______
```

7: Manager wants to delete a list? Task? Board?

```
    DELETE FROM Task
    WHERE taskTitle = _____;
```

8: Changing the name of a list/task/board

```
    UPDATE ProjectBoard
    SET projectTitle = newTitle
    WHERE pID = __ ;

    UPDATE Lists
    SET listTitle= newTitle
    WHERE lID = __ ;

    UPDATE Tasks
    SET taskName = newName
    WHERE tID = __;
```

1. Employee wants to see the creation dates of the project boards they're in

```
    Select P.CreateDate, P.ProjectTitle
```

```
FROM ProjectBoard P, WorksOn W, Employee E,
WHERE W.pID = P.pID, W.eID = E.eID
```

2. Employee’s permissions are updated to that of a manager’s

```
UPDATE Employee
SET permissions = (manager?)
WHERE Ename = __ ;
```

3. An employee quits, so they’re removed from all project boards they’re on.

```
DELETE FROM Employee
WHERE eID = __ ;
```

4. Manager wants to see all project boards in existence

```
SELECT projectTitle
FROM ProjectBoard
ORDER BY projectTitle
```

5. List of all project boards an employee is a member of

```
SELECT P.projectTitle
FROM WorksOn W, Employee E, ProjectBoard P,
WHERE W.eID = E.eID AND W.pID = P.pID;
```

6. The number of project boards an employee is a member of

```
SELECT count(W.pID)
FROM WorksOn W, Employee E,
WHERE W.eID = E.eID and E.eID = __;
```

Example Data Tables:

Employee:

Eid	**Name**	**PermID**
1001	Devon Kenzie	001
1002	Jenny Huang	002
1003	Jamie Polintan	003
0001	Patrice Belleville	101
0002	Gregor Kiczales	102

Manager:

Eid	Name	PositionTitle
0001	Patrice Belleville	Manager
0002	Gregor Kiczales	Manager
0003	Celina Berg	Manager
0004	Cinda Heeren	Manager
0005	Paul Carter	Manager

RegularWorker:

Eid	Name
1001	Devon Kenzie
1002	Jenny Huang
1003	Jamie Polintan
0001	Braedyn Kenzie
0002	Jason Xun

WorksOn:

pID	eID
2001	1001
2002	1002
2003	0004
2004	1003
2005	0005

ProjectBoard:

pID	DateCreated	Title
2001	5/30/2015 9:15	Midterm
2002	5/23/2015 9:02	ProjectProposal
2003	5/23/2015 9:04	ERDiagramsAndSchemas
2004	5/26/2015 17:36	ProjectFormalSpecification
2005	6/12/2015 12:46	IndividualReport

List:

pID	ListID	ListTitle
2001	3001	ToDo
2001	3002	Done
2001	3003	Brainstorm
2002	3001	ToDo
2002	3002	Done

Task:

pID	ListID	TaskID	TaskTitle
2001	3001	4001	CoverPage
2001	3002	4002	SQLDDL
2001	3002	4001	CoverPage
2002	3003	4003	Schema
2002	3003	4004	ERDiagram

Permission:

eID	permID	canCreate	canDelete
1001	0001	0	0
1002	0002	0	0
1003	0003	0	0
0001	0101	1	1
0002	0102	1	1

Employee(eID: int, name: string, **permID**: int)
Manager(**eID**:int, Name:string, positionTitle:string)
RegularW(**eID**:int, Name:string)
WorksOn(**pID**:int, **eID**:int)
ProjectBoard(ProjID:int, Title:string, DateCreated:datetime)
List(**pid**:int, ListID:int, ListTitle:string)
Task(**pID**:int, **ListID**:int, TaskID:int, taskTitle:string)
Permissions(**eID**:int, permID:int, canCreate:Boolean, canDelete:Boolean)

***foreign key**
*primary key