IV. Document the final term project report.

a) Problem description

For this project we were asked to draw an EER diagram that met a set of requirements laid out by a company. After the initial draft we had to take the EER, perform relational mapping, normalize it to 3NF, and change the EER to display the changes made. After that we needed to code the database and run various queries.

b) Individual contribution breakdown

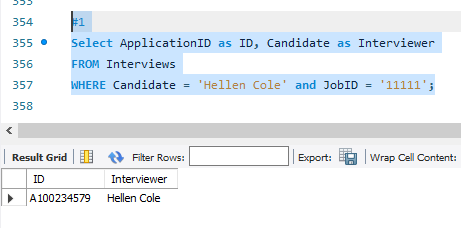
Hyungtaek: EER Diagram, Creating Tables, Views, Tuple Insert

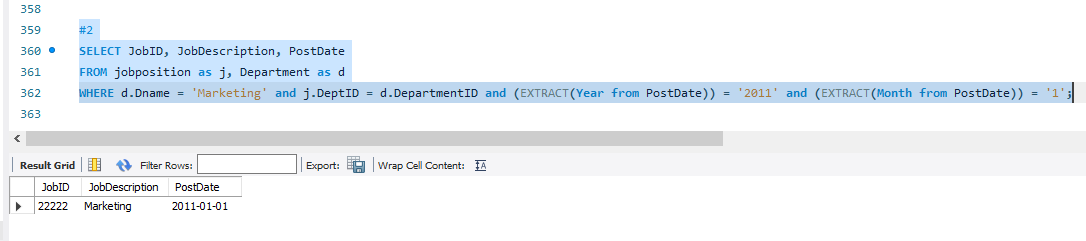
Geon: EER Diagram, SQL statements and Queries, Edit Table

Wyatt: EER Diagram, Relational Mapping, normalizing to 3NF, Paper, Contributed & Finalized Queries

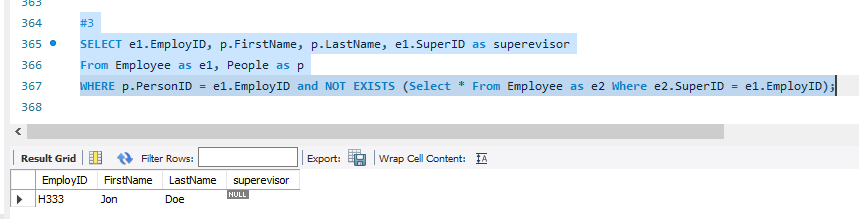
c) Project questions (Answer questions listed in this project).

1. Return the ID and Name of interviewers who participate in interviews where the interviewee's name is "Hellen Cole" arranged for job "11111".



2. Return the ID of all jobs which are posted by department "Marketing" in January 2011. 

3. Return the ID and Name of the employees having no supervisees.

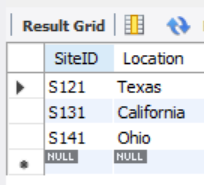


4. Return the Id and Location of the marketing sites with no sale records during March 2011.

SELECT SiteID, Location

FROM Site as s

WHERE NOT EXISTS (Select \* from purchaseatsite as p Where s.SiteID = p.SiteID and (EXTRACT(Year from p.SaleTime)) = '2011' and (EXTRACT(Month from p.SaleTime)) = '3');

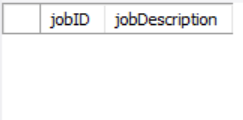


5. Return the job's id and description, which does not hire a suitable person one month after it is posted.

SELECT j.jobID, j.jobDescription

from jobPosition as j, Interviews as i

Where j.JobID = i.JobID and (EXTRACT(Month from j.PostDate) - EXTRACT(Month from i.InterviewDate)) > 1 and (EXTRACT(Year from j.PostDate) - EXTRACT(Year from i.InterviewDate)) >= 0;



6. Return the ID and Name of the salespeople who have sold all product types whose price is above $200.

SELECT distinct(pur.EmployID), p.FirstName, p.LastName

FROM purchaseatsite pur, Employee e, People p, Product prod

WHERE pur.EmployID = e.EmployID AND e.employID= p.PersonID and prod.ListPrice > 200;

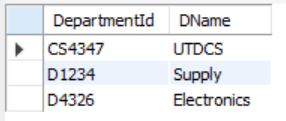


7. Return the department's id and name, which has no job post during 1/1/2011 and 2/1/2011.

SELECT DepartmentId, DName

From department as d

WHERE NOT EXISTS(Select d.DepartmentID From JobPosition as j where d.departmentID = j.deptID and EXTRACT(Month from j.PostDate) = '1' and EXTRACT(Year from j.PostDate) = '2011' ) AND NOT EXISTS(Select d.DepartmentID From JobPosition as j where d.departmentID = j.deptID and EXTRACT(Month from j.PostDate) = '2' and EXTRACT(Year from j.PostDate) = '2011' );

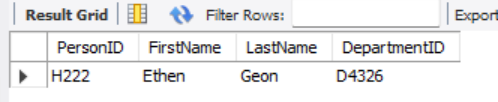


8. Return the ID, Name, and Department ID of the existing employees who apply for job "12345".

SELECT distinct(p.PersonID), p.FirstName, p.LastName, e.DepartmentID

FROM People as P, Employee as e, Department as D, jobposition as JP, Interviews as I

Where P.PersonID = E.EmployID and E.ApplicationID = I.ApplicationID and I.JobID ='12345';



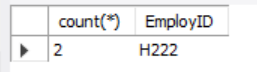
9. Return the best seller's type in the company (sold the most items).

select count(\*), EmployID

from purchaseatsite as p

group by EmployID order by count(\*) desc

limit 1;



10. Return the product type whose net profit is highest in the company (money earned minus the part cost).

SELECT ProductType from PRODUCT

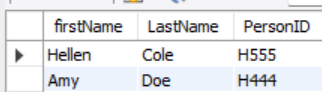
HAVING max(ListPrice);



11. Return the name and id of the employees who have worked in all departments after being hired by the company.

SELECT firstName, LastName, peo.PersonID FROM People as peo, PotentialEmployee pot

WHERE peo.PersonID = pot.PersonID;

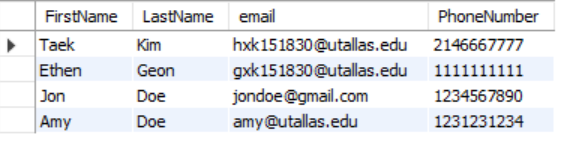


12. Return the name and email address of the interviewee who is selected.

SELECT distinct(FirstName), LastName, email

FROM People, employee, application, interviews

WHERE people.PersonID = Employee.EmployID and interviews.applicationID = employee.applicationID;



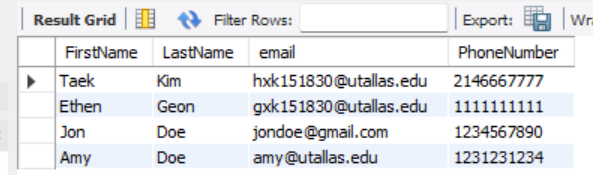
13. Retrieve the names, phone numbers, and email addresses of the interviewees selected for all the jobs they apply for.

SELECT distinct(PEOPLE.FirstName), People.LastName, People.email, PersonPhone.PhoneNumber

FROM PEOPLE, PersonPhone, potentialemployee, application

WHERE potentialemployee.ApplicationID = application.ApplicationID AND PersonPhone.PersonID = People.PersonID

GROUP BY FirstName;



14. Return the employee's name and id whose average monthly salary is the highest in the company.

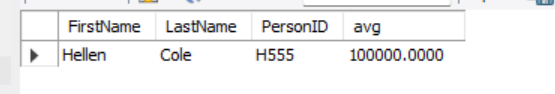
SELECT FirstName, LastName, PersonID, avg(pay.amount) as avg

FROM People,Pay

GROUP BY pay.amount

ORDER BY avg desc

LIMIT 1;



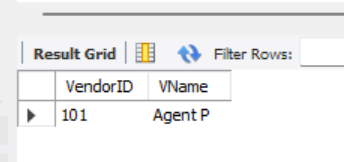
15. Return the ID and Name of the vendor who supplies part whose name is "Cup" and weight is smaller than 4 pounds, and the price is lowest among all vendors.

SELECT V.VendorID,V.VName

FROM VENDOR V,parts as p,product

WHERE p.PartName = 'CUP' AND p.WEIGHT<4

HAVING MIN(PRODUCT.ListPrice);



d) EER diagram with all assumptions.

* Final 3NF EER from folder

e) Relation schema after normalization. All relations must be in 3NF. The relation schema should include primary and foreign keys (if any) for all relations.

* 3NF\_Relational\_map.pdf from folder

f) All requested SQL statements.

* Project\_cs4347\_update\_final.sql from folder

g) Dependency diagram.

* Functional\_Dependency.pdf from folder