*Kyoto Computer Gakuin – Advanced Topics in Database Technologies Roberto Espinoza –* [*r\_espinoza@kcg.edu*](mailto:r_espinoza@kcg.edu)

Advanced Topics in Database Technologies

– Practical Assignment – Final Review –

1. Considering the COMPANY database, answer the items below:
   1. Write down the SQL command and the **relational calculus** expression for a query that returns all the data from the employee, and the department name and manager name of their department.
      * SELECT employee. \*, department.dname, CONCAT(manager.fname, ' ', manager.lname) AS manager\_name FROM employee

JOIN department ON employee.dno = department.dnumber

JOIN employee AS manager ON department.mgr\_ssn = manager.ssn;

* + - {e, d.dname, m.fname, m.lname | employee(e)

AND department(d) AND employee(m)

AND (e.dno = d.dnumber)

AND (d.mgr\_ssn = m.ssn)

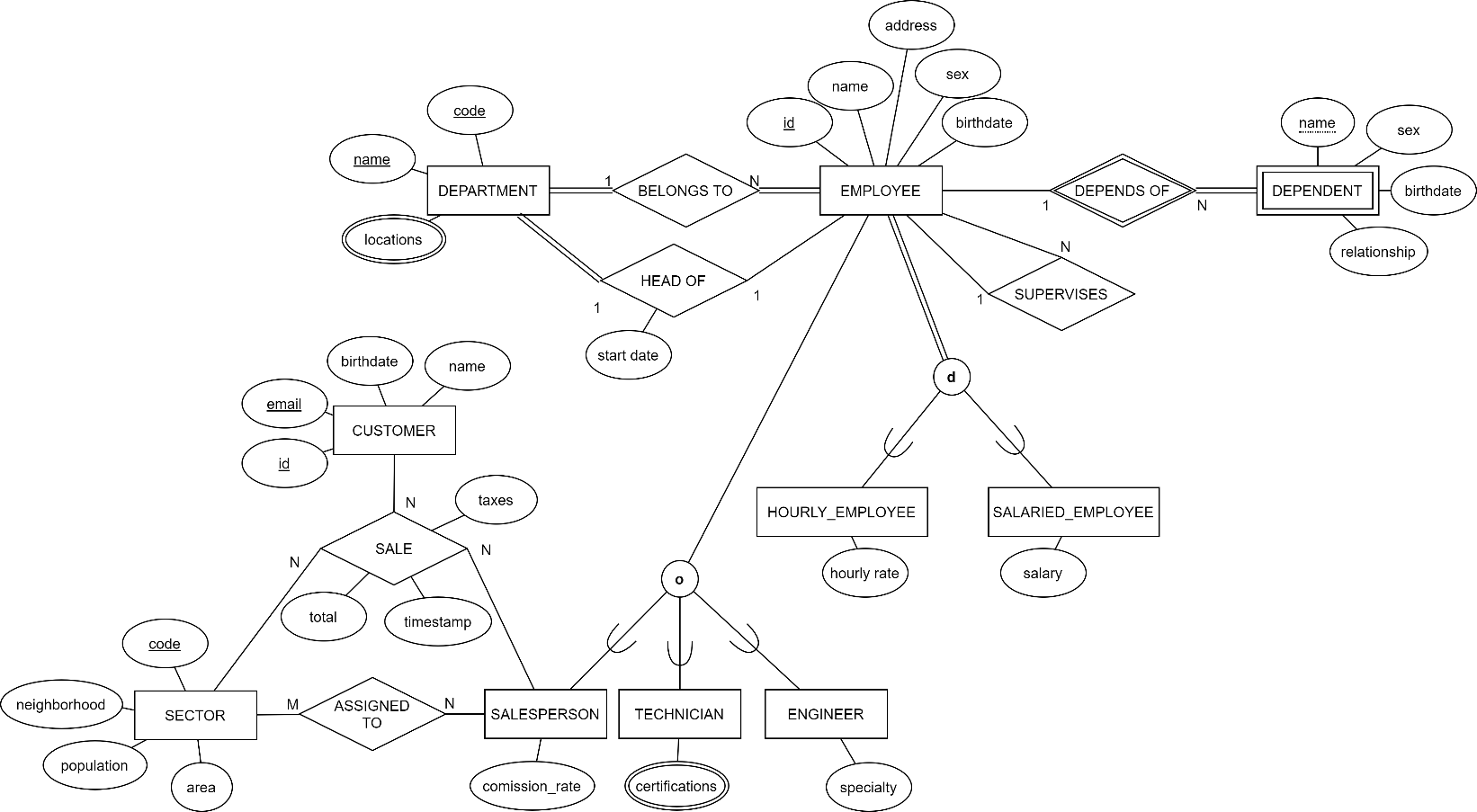
}

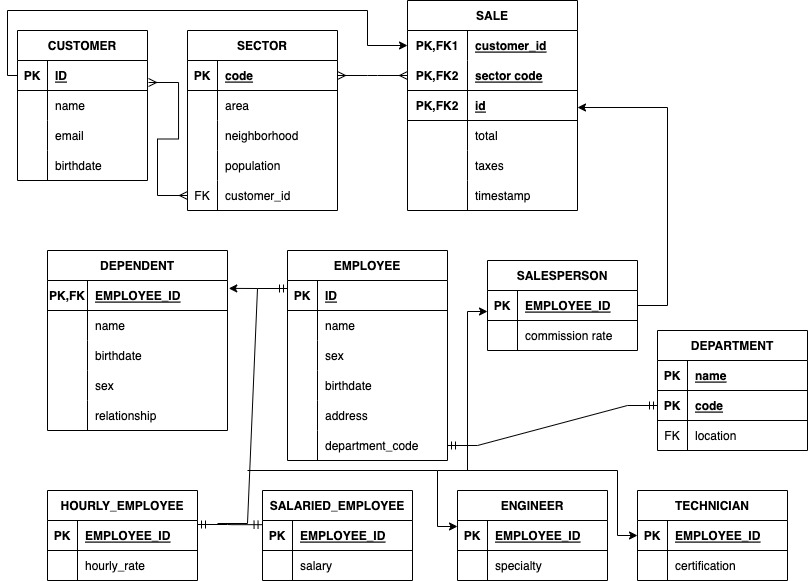
* 1. Write down the SQL command and the **relational algebra** expression for a query that returns all attributes for employees that have at least 2 subordinates.
     + SELECT e. \* FROM employee e WHERE (SELECT COUNT (\*) FROM employee sub

WHERE sub.super\_ssn = e.ssn) >= 2

* + - π e. \* (σ (sub.super\_ssn = e.ssn) (employee sub) ⋅ σ (COUNT (\*) >= 2) (employee))

1. Follow the procedure learned in class to map the EERM below to a Relational Model Diagram and show the final diagram.





1. Write down a regular expression that matches:
   1. Valid hexadecimal number literals. Consider that a hexadecimal number literal is defined a sequence started with either “0x”, or “0X”, or “h”, or “H”, followed by one or more digits or letters from A to F (upper or lower case).
      * ^(0x|0X|h|H)[0-9a-fA-F]+$
   2. A person’s name. Each part of the name must start with an uppercase letter. The middle parts can be abbreviated as one uppercase letter followed by a dot. The first and last parts of the name cannot be abbreviated.
      * ^[A-Z][a-z]+(?:[.][A-Z][a-z]+)? [A-Z][a-z]+$
2. Considering the COMPANY database, write an SQL expression that returns the name of the manager and the number of employees in the department, for departments that have at least 2 employees:
   1. Using a join
      * SELECT department.dname, COUNT(employee.ssn) as num\_employees,

CONCAT (manager.fname, ' ', manager.lname) AS manager\_name

FROM employee

JOIN department ON employee.dno = department.dnumber

JOIN employee AS manager ON department.mgr\_ssn = manager.ssn

GROUP BY department.dname, department.dnumber, CONCAT(manager.fname, ' ', manager.lname)

HAVING COUNT (employee.ssn) >= 2;

* 1. Using a nested query
     + SELECT department.dname, COUNT(employee.ssn) as num\_employees, CONCAT(manager.fname, ' ', manager.lname) AS manager\_name

FROM employee

JOIN department ON employee.dno = department.dnumber

JOIN employee AS manager ON department.mgr\_ssn = manager.ssn

WHERE department.dnumber IN (SELECT dno

FROM employee GROUP BY dno HAVING COUNT (ssn) >= 2)

GROUP BY department.dname, department.dnumber, CONCAT(manager.fname, ' ', manager.lname);

1. Considering the COMPANY database:
   1. Create a trigger that, whenever an employee is inserted or updated, makes sure that this employee works in at least 1 project.
      * CREATE FUNCTION process\_employee() RETURNS TRIGGER

AS $$

BEGIN

IF NOT EXISTS (SELECT \* FROM works\_on WHERE essn = NEW.ssn) THEN

RAISE EXCEPTION 'Employee must work in at least 1 project';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER process\_empl\_trigger

AFTER INSERT OR UPDATE ON employee

FOR EACH ROW

EXECUTE FUNCTION process\_employee();

* 1. Create a CHECK constraint that achieves the same result.
     + ALTER TABLE employee ADD CONSTRAINT check\_employee\_works\_on

CHECK (EXISTS (SELECT \* FROM works\_on WHERE essn = NEW.ssn));