INFO20003 Database Systems

Week 5

Relational Algebra

- a collection of operators, which take instance(s) of a relation as **operand**(s), and returns a relation instance as an **output**

- An operator can be either unary or binary

Fundamental Operations

5 basic operators for Relational Algebra:

- Selection (σ)
- Projection (π)
- Set-difference $(-)$
- Union $(\cup)$
- Cross Product $(\times)$

Removal Operators

- Projection: removes columns
- Selection: removes rows

Projection

$$\pi_{A1, A2, \ldots, An}(R)$$

- R: input relation
- A1, A2, …, An: attributes that are projected

- Output: create a new relation with the projected attributes

Projection

FirstName	LastName	Phone	Email
Jon	Snow	0551-999-210	knowsnothing@hotmail.com
Daenerys	Targaryen	0569-988-112	bendtheknee@gmail.com
Jamie	Lannister	0531-987-654	handsfree@gmail.com
Night	King	0566-123-456	killerstare@gmail.com

The expression $\pi_{\text{FirstName, LastName}}$ (Person) will result in:

FirstName	LastName
Jon	Snow
Daenerys	Targaryen
Jamie	Lannister
Night	King

Selection

$\sigma_C(\mathrm{R})$

- R: input relation
- C: condition used to filter rows

- Output: creates a new relation consisting of the rows for which C is true

Selection

FirstName	LastName	Phone	Email
Jon	Snow	0551-999-210	knowsnothing@hotmail.com
Daenerys	Targaryen	0569-988-112	bendtheknee@gmail.com
Jamie	Lannister	0531-987-654	handsfree@gmail.com
Night	King	0566-123-456	killerstare@gmail.com

$$\sigma_{\text{FirstName} = \text{'Jon'} \vee \text{LastName} = \text{'King'}} (\text{Person})$$

FirstName	LastName	Phone	Email
Jon	Snow	0551-999-210	knowsnothing@hotmail.com
Night	King	0566-123-456	killerstare@gmail.com

Combine expressions

$$\pi_{\text{FirstName, LastName}}(\sigma_{\text{FirstName} = \text{'Jon'} \vee \text{LastName} = \text{'King'}}(\text{Person}))$$

FirstName	LastName
Jon	Snow
Night	King

Set Operators

- Set-difference
- Union

- Input relations R and S must have the **same attributes** with the **same domains**, in the **same order**

Set-Difference

R – S

- R & S: 2 relations with the same attributes
- Output: every row which is in R but not in S

RandomCombo1

FirstName	LastName
Jon	Snow
Daenerys	Targaryen
Jamie	Lannister
Night	King

RandomCombo2

FirstName	LastName
Night	King
Arya	Stark
Cersei	Lannister
Daenerys	Targaryen

RandomCombo1 – RandomCombo2 will result in:

FirstName	LastName
Jon	Snow
Jamie	Lannister

Union

$R \cup S$

- R&S: 2 relations with the same attributes
- Output: every row which is in either R or S

GoodGuys

FirstName	LastName
Jon	Snow
Daenerys	Targaryen

BadGuys

FirstName	LastName
Cersei	Lannister
Night	King

GoodGuys ∪ BadGuys will result in:

FirstName	LastName
Jon	Snow
Daenerys	Targaryen
Cersei	Lannister
Night	King

Cross Product

$R \times S$

- Each row of R pairs with **each** row of S
- The resulting schema has **all the attributes** from both relations
- If some attributes have **same name**, **rename** them by using renaming operator

Cross Product

Person

FirstName	LastName	Email
Jon	Snow	knowsnothing@hotmail.com
Night	King	killerstare@gmail.com

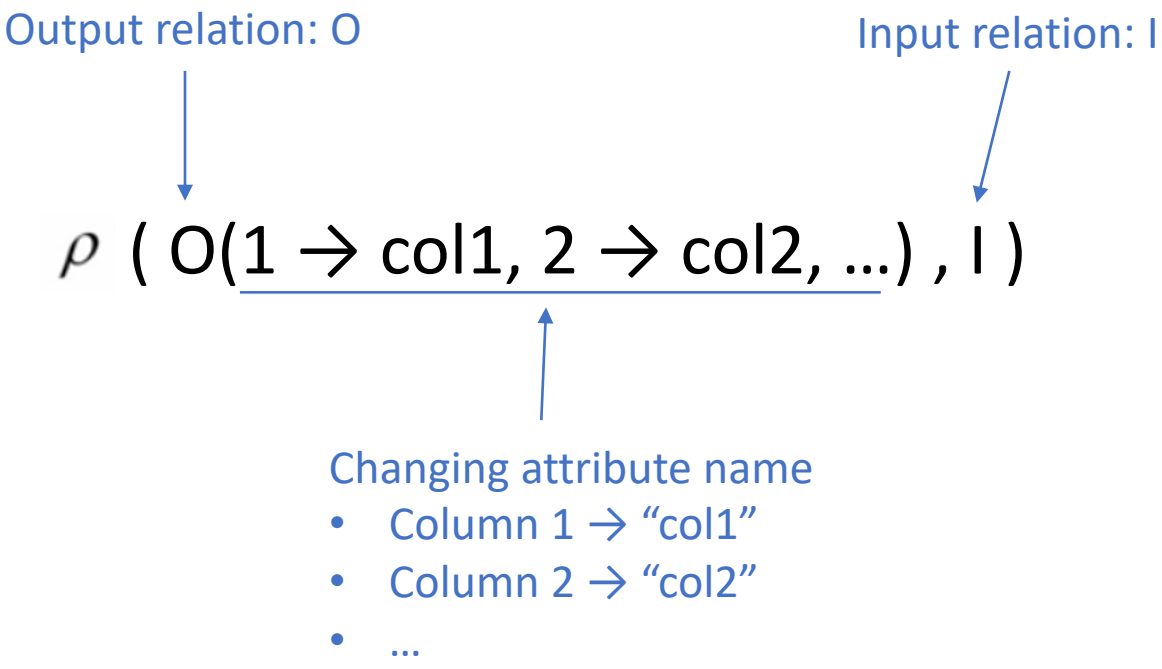
Weapon

Weapon	Metal
Sword	Valyrian steel
Dagger	Dragon glass

Person × Weapon will result in:

FirstName	LastName	Email	Weapon	Metal
Jon	Snow	knowsnothing@hotmail.com	Sword	Valyrian steel
Jon	Snow	knowsnothing@hotmail.com	Dagger	Dragon glass
Night	King	killerstare@gmail.com	Sword	Valyrian steel
Night	King	killerstare@gmail.com	Dagger	Dragon glass

Rename


Output relation: O
Input relation: I
ρ (O(1 → col1, 2 → col2, …) , I)
Changing attribute name
• Column 1 → “col1”
• Column 2 → “col2”
• …

Rename example

$$\rho(S1(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow rating1, 4 \rightarrow age1), Sailors)$$

Compound Operations

- Intersection
- Natural Join
- Condition join (Theta/Inner join)

Intersection

$$R \cap S = R - (R - S)$$

- Output: a relation containing all the tuples which are present in both relations

RandomCombo1

FirstName	LastName
Jon	Snow
Daenerys	Targaryen
Jamie	Lannister
Night	King

RandomCombo2

FirstName	LastName
Night	King
Arya	Stark
Cersei	Lannister
Daenerys	Targaryen

RandomCombo1 ∩ RandomCombo2 will result in:

FirstName	LastName
Daenerys	Targaryen
Night	King

Compound Operations

- Intersection
- Natural Join
- Condition join (Theta/Inner join)

Natural Join

$R \bowtie S$

- Create a new relation, pairing each tuple from R and S where the **common attributes are equal**

A natural join can be broken down into following steps:

- Compute $R \times S$
- Select rows where attributes that appear in both relations have equal values.
- Project all unique attributes and one copy of each of the common ones.

Natural Join example

Person

FirstName	LastName	Email
Jon	Snow	knowsnothing@hotmail.com
Daenerys	Targaryen	bendtheknee@gmail.com
Tyrion	Lannister	idrinkandiknow@gmail.com
Night	King	killerstare@gmail.com

WeaponOwner

Weapon	LastName	Metal
Sword	Snow	Valyrian steel
Dagger	Lannister	Dragon glass

Common attribute = LastName

Natural Join example

Person × Weapon (intermediate result):

FirstName	LastName	Email	Weapon	LastName	Metal
Jon	Snow	knowsnothing@hotmail.com	Sword	Snow	Valyrian steel
Jon	Snow	knowsnothing@hotmail.com	Dagger	Lannister	Dragon glass
Daenerys	Targaryen	bendtheknee@gmail.com	Sword	Snow	Valyrian steel
Daenerys	Targaryen	bendtheknee@gmail.com	Dagger	Lannister	Dragon glass
Tyrion	Lannister	idrinkandiknow@gmail.com	Sword	Snow	Valyrian steel
Tyrion	Lannister	idrinkandiknow@gmail.com	Dagger	Lannister	Dragon glass
Night	King	killerstare@gmail.com	Sword	Snow	Valyrian steel
Night	King	killerstare@gmail.com	Dagger	Lannister	Dragon glass

Natural Join example

Person ⋈ Weapon will result in:

FirstName	LastName	Email	Weapon	Metal
Jon	Snow	knowsnothing@hotmail.com	Sword	Valyrian steel
Tyrion	Lannister	idrinkandiknow@gmail.com	Dagger	Dragon glass

Compound Operations

- Intersection
- Natural Join
- Condition join (Theta/Inner join)

Conditional Join

$R \bowtie_C S$

- joins rows from relation R and S such that the Boolean condition C is true

- $R \bowtie_C S = \sigma_C(R \times S)$

Conditional Join example

Person

FirstName	LastName	Email
Jon	Snow	knowsnothing@hotmail.com
Daenerys	Targaryen	bendtheknee@gmail.com
Tyrion	Lannister	idrinkandiknow@gmail.com
Night	King	killerstare@gmail.com

WeaponOwner

Weapon	Name	Metal
Sword	Snow	Valyrian steel
Dagger	Lannister	Dragon glass

$$\textbf{Person} \bowtie_{\text{LastName} = \text{Name}} \textbf{WeaponOwner}$$

Conditional Join example

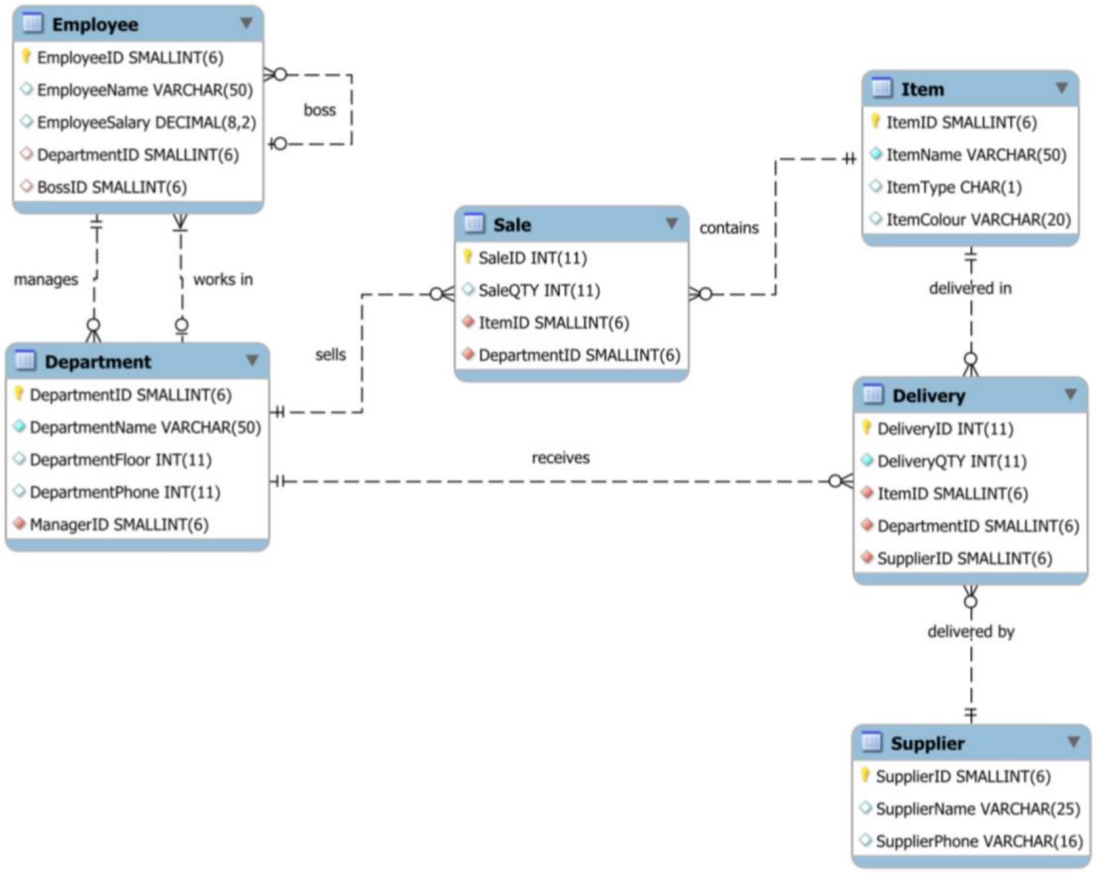
Person × Weapon (intermediate result):

FirstName	LastName	Email	Weapon	Name	Metal
Jon	Snow	knowsnothing@hotmail.com	Sword	Snow	Valyrian steel
Jon	Snow	knowsnothing@hotmail.com	Dagger	Lannister	Dragon glass
Daenerys	Targaryen	bendtheknee@gmail.com	Sword	Snow	Valyrian steel
Daenerys	Targaryen	bendtheknee@gmail.com	Dagger	Lannister	Dragon glass
Tyrion	Lannister	idrinkandiknow@gmail.com	Sword	Snow	Valyrian steel
Tyrion	Lannister	idrinkandiknow@gmail.com	Dagger	Lannister	Dragon glass
Night	King	killerstare@gmail.com	Sword	Snow	Valyrian steel
Night	King	killerstare@gmail.com	Dagger	Lannister	Dragon glass

Person $\bowtie_{\text{LastName} = \text{Name}}$ **Weapon**

FirstName	LastName	Email	Weapon	Name	Metal
Jon	Snow	knowsnothing@hotmail.com	Sword	Snow	Valyrian steel
Tyrion	Lannister	idrinkandiknow@gmail.com	Dagger	Lannister	Dragon glass

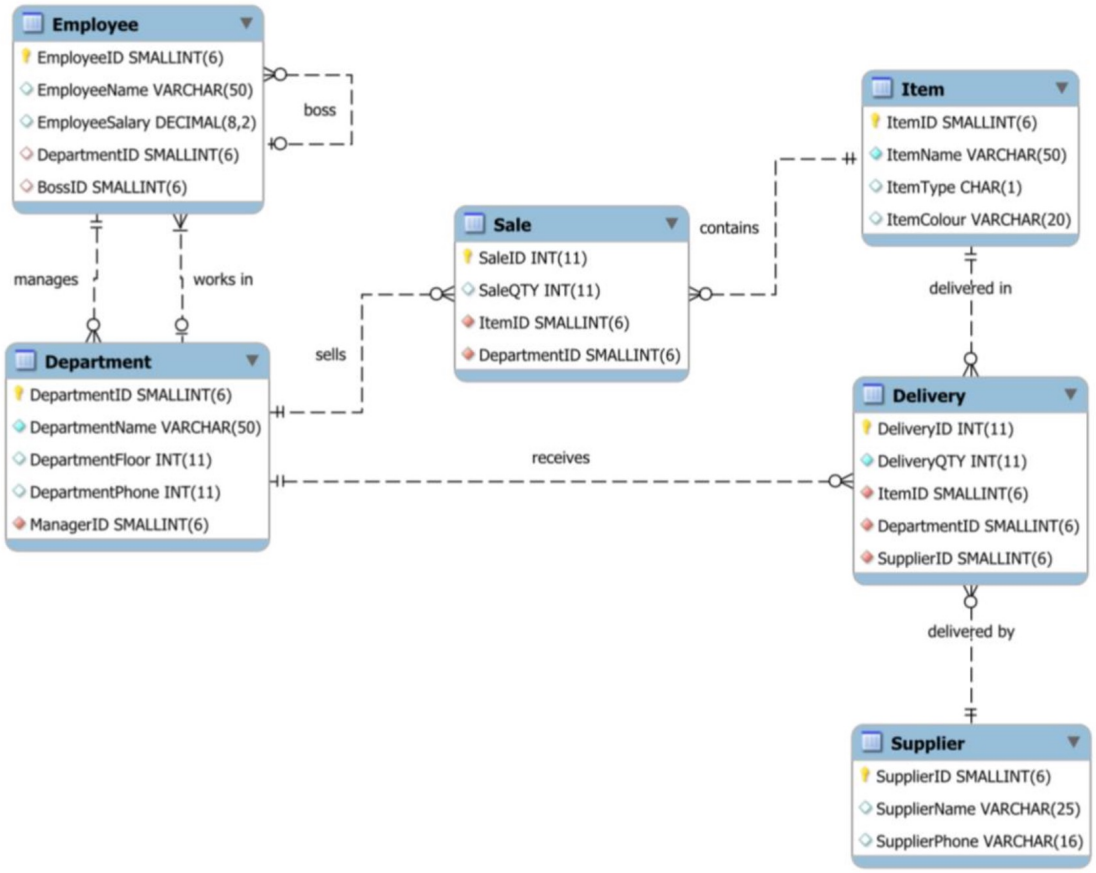
Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2a)

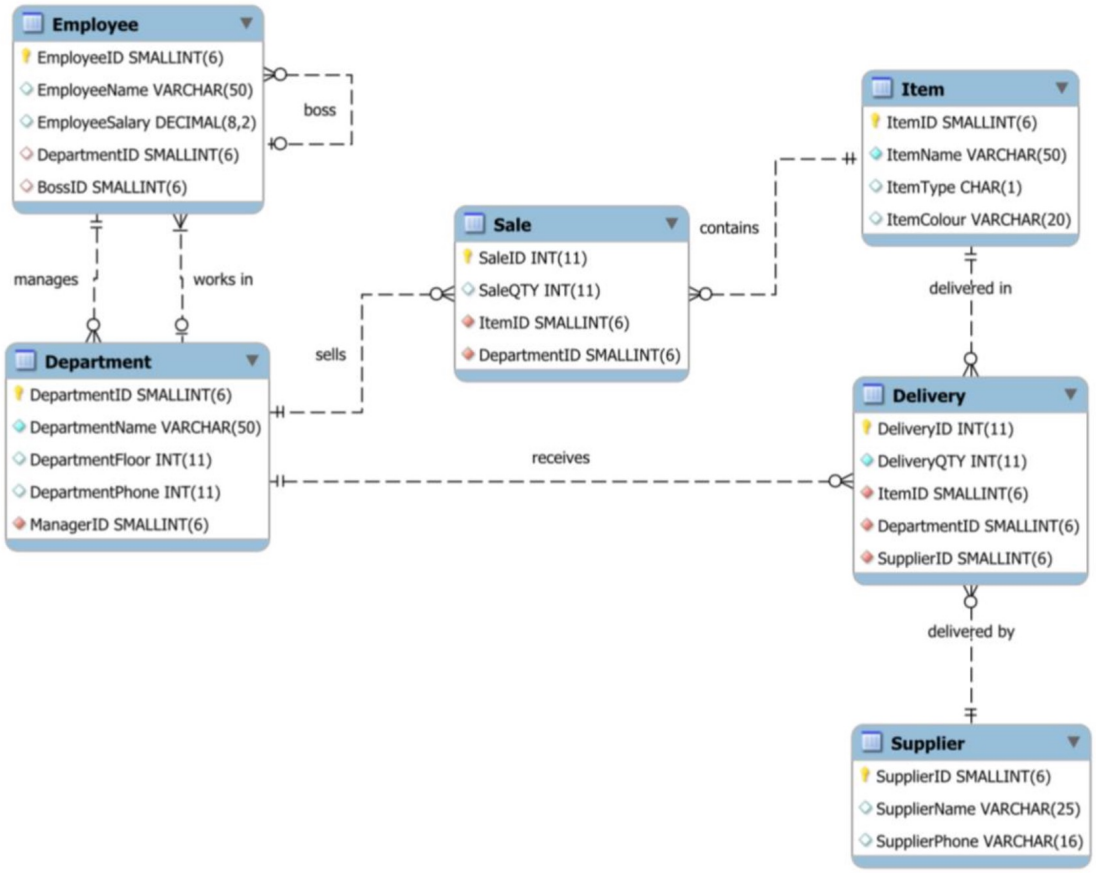
a. Find the names of all employees.

Relational Algebra: $\pi_{\text{EmployeeName}}(\text{Employee})$

SQL:

```
SELECT EmployeeName
FROM Employee;
```

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2b)

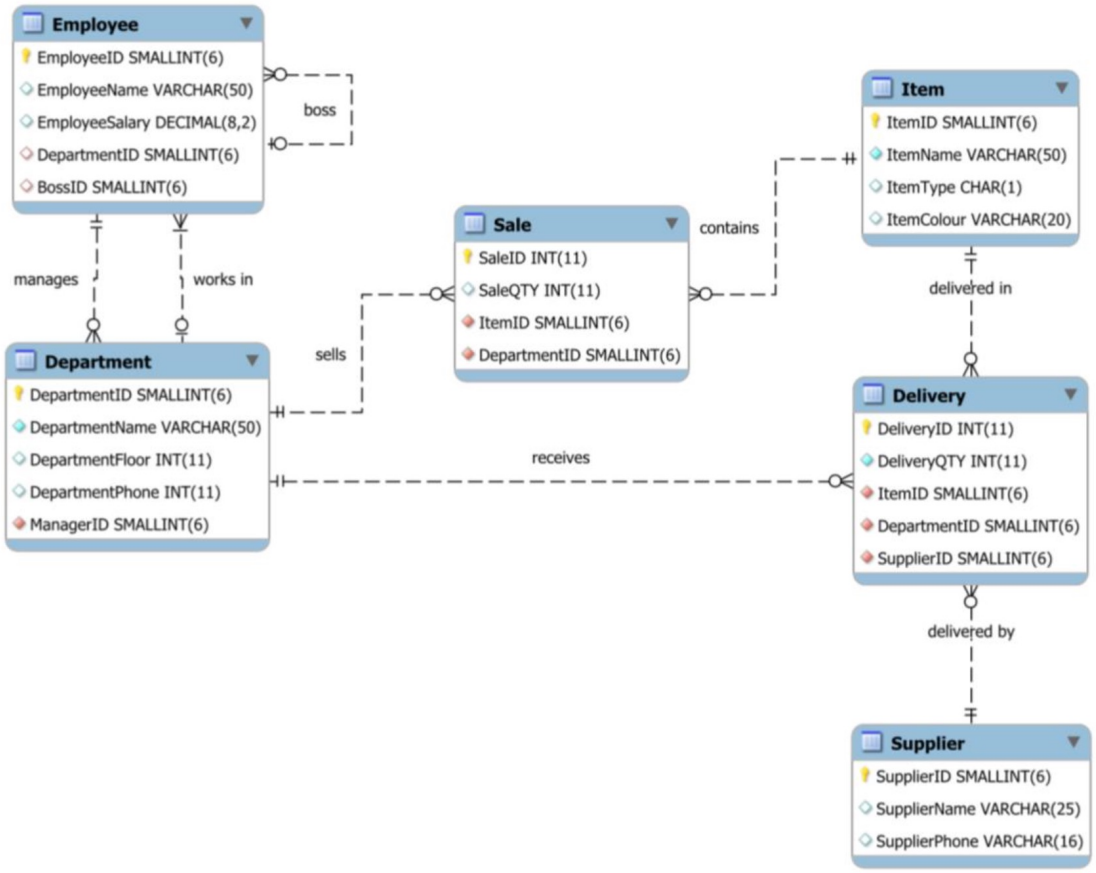
b. Find the names of all employees in department number 1.

Relational Algebra: $\pi_{\text{EmployeeName}}(\sigma_{\text{DepartmentID}=1}(\text{Employee}))$

SQL:

```
SELECT EmployeeName
FROM Employee
WHERE DepartmentID = 1;
```

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2c)

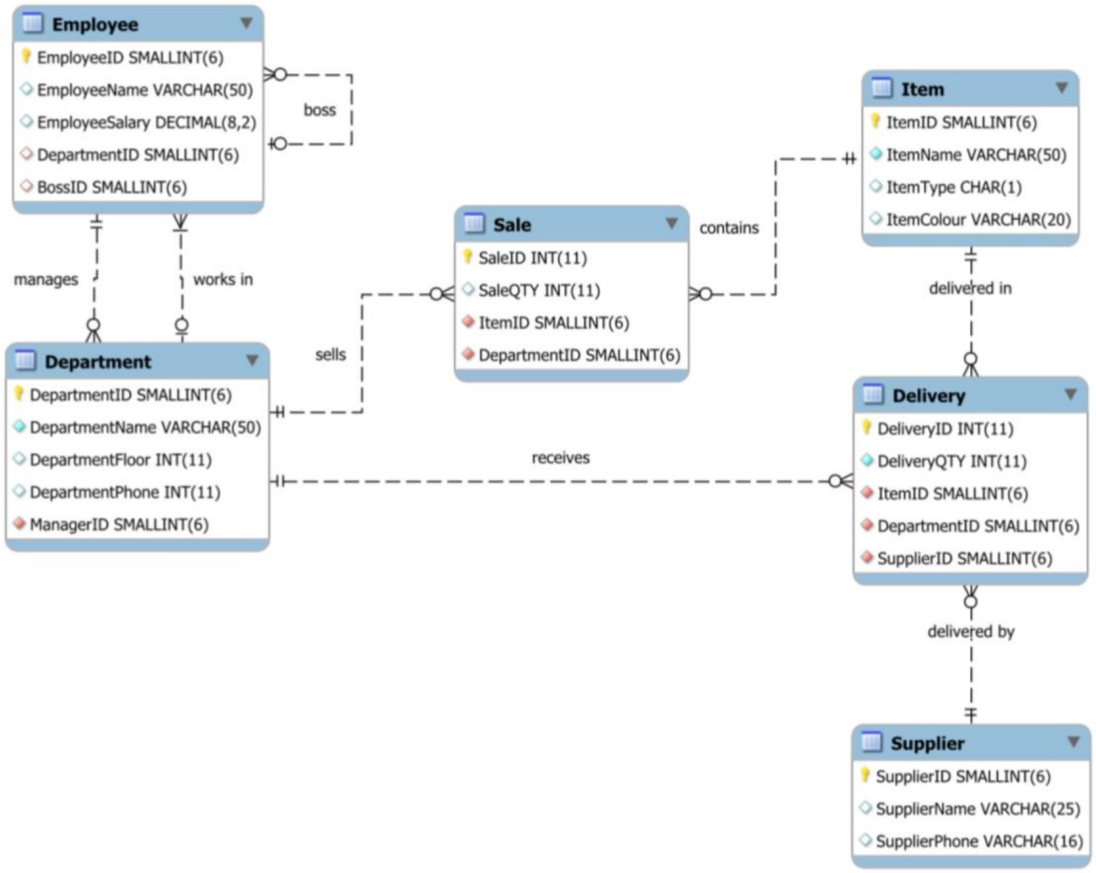
c. List the names of green items of type C.

Relational Algebra: $\pi_{\text{ItemName}}(\sigma_{\text{ItemColour} = \text{'Green'} \wedge \text{ItemType} = \text{'C'}}(\text{Item}))$

SQL:

```
SELECT ItemName
FROM Item
WHERE ItemType = 'C' AND ItemColour = 'Green';
```

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2d)

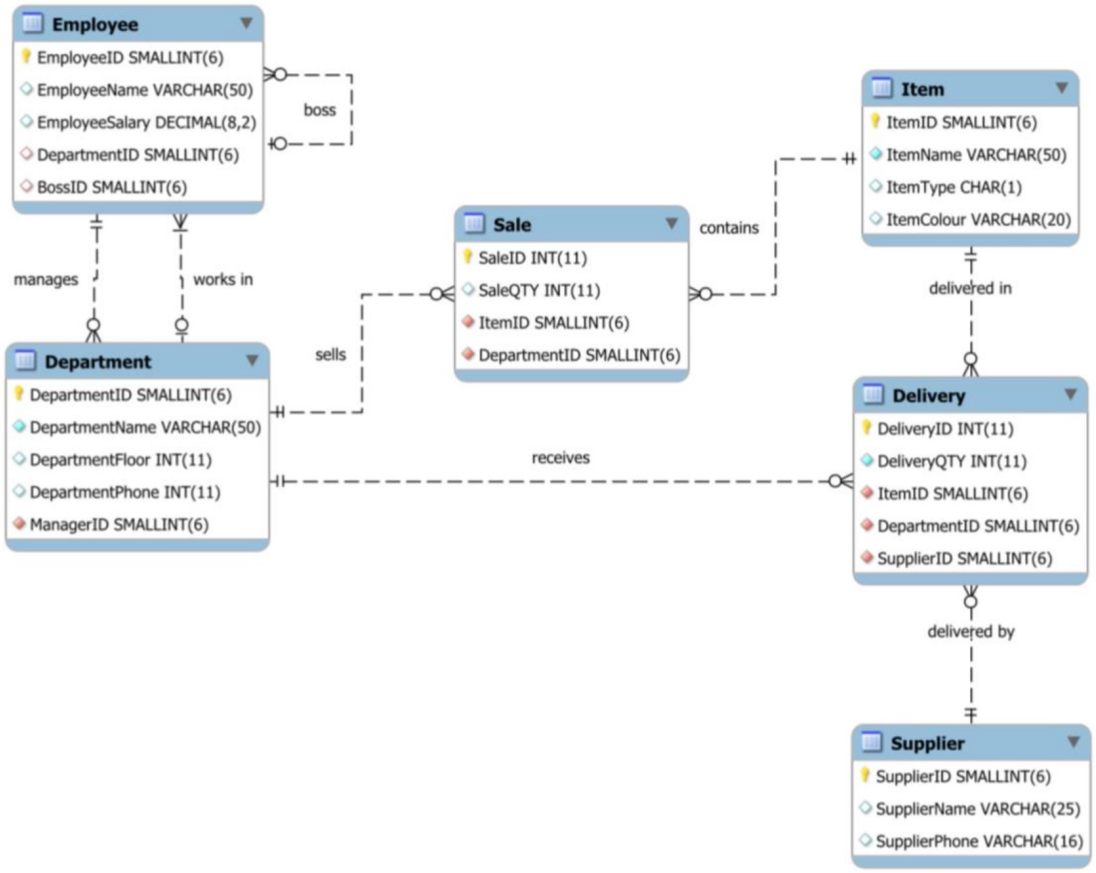
d. Find the items sold by the departments on the second floor (only show ItemID).

Relational Algebra: $\pi_{\text{ItemID}}(\sigma_{\text{DepartmentFloor}=2}(\text{Sale} \bowtie \text{Department}))$

SQL:

```
SELECT DISTINCT ItemID
FROM Sale NATURAL JOIN Department
WHERE DepartmentFloor = 2;
```

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2e)

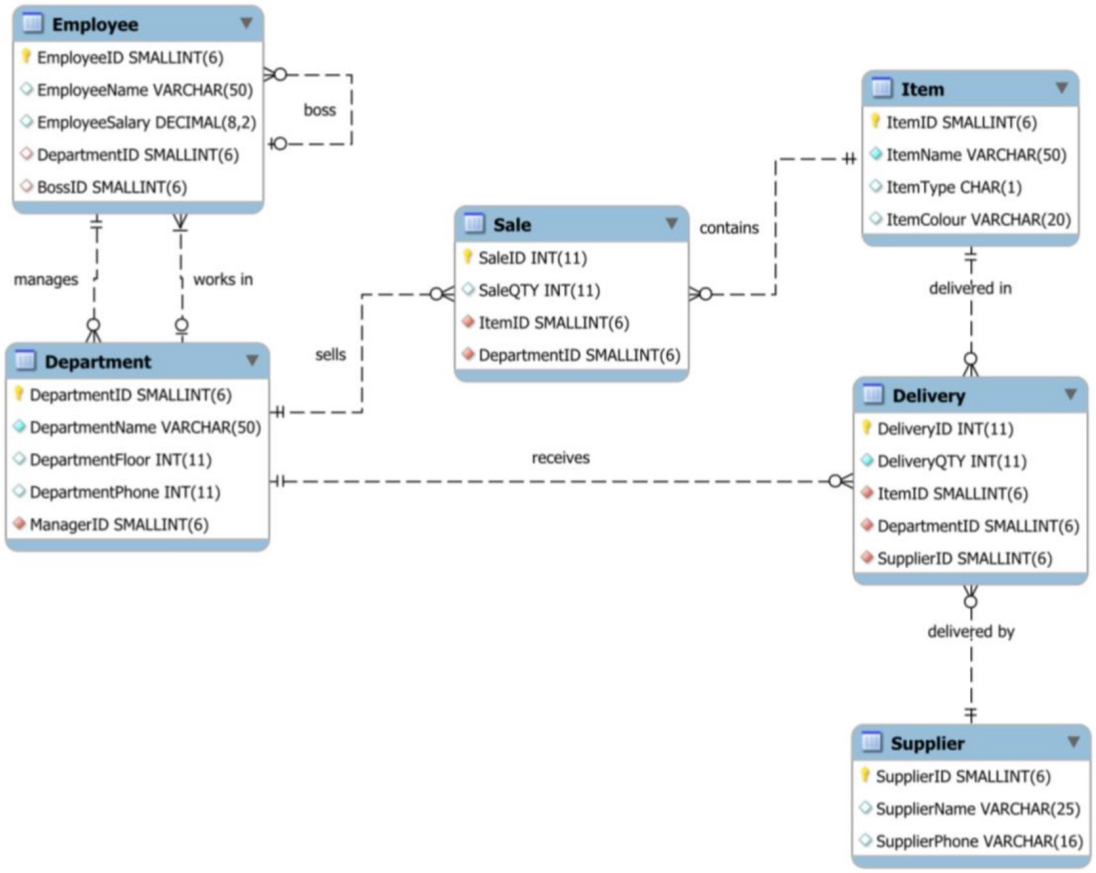
e. Find the names of brown items sold by the Recreation department.

Relational Algebra: $\pi_{\text{ItemName}}(\sigma_{\text{DepartmentName} = \text{'Recreation'} \wedge \text{ItemColour} = \text{'Brown'}}(\text{Item} \bowtie \text{Sale} \bowtie \text{Department}))$

SQL:

```
SELECT ItemName
FROM Item NATURAL JOIN Sale NATURAL JOIN Department
WHERE DepartmentName = 'Recreation'
  AND ItemColour = 'Brown';
```

Q2)



Solve the following problems using relational algebra (RA) and translate to SQL statements:

a) Find the names of all employees.
b) Find the names of all employees in department with id of 1.
c) List the names of green items of type C.
d) Find the items sold by the departments on the second floor (only show ItemID)
e) Find the names of brown items sold by the Recreation department.
f) Find the employees whose salary is less than half that of their boss

Q2f)

f. Find the employees whose salary is less than half that of their managers.

Below are two examples using the rename (ρ) operator:

ρ(Emp(EmployeeName → EmpName, EmployeeSalary → EmpSalary, BossID → EmpBossID), Employee) ρ(Boss(EmployeeID → BossEmployeeID, EmployeeSalary → BossSalary), Employee) $\pi_{\text{EmpName}}(\sigma_{\text{EmpSalary} < (\text{BossSalary}/2)}(\text{Emp} \bowtie_{\text{EmpBossID} = \text{BossEmployeeID}} \text{Boss}))$	ρ(Boss(BossID → BossBossId, EmployeeID → BossID, EmployeeSalary → BossSalary, DepartmentID → BossDepID, EmployeeName → BossName), Employee) $\pi_{\text{EmployeeName}}(\sigma_{\text{EmployeeSalary} < (\text{BossSalary}/2)}(\text{Employee} \bowtie \text{Boss}))$

Or you could use an SQL-like notation:

Emp := Employee

Boss := Employee

$\pi_{\text{Emp.EmployeeName}}(\sigma_{\text{Emp.EmployeeSalary} < (\text{Boss.EmployeeSalary}/2)}($
Emp $\bowtie_{\text{Emp.BossID} = \text{Boss.EmployeeID}}$ Boss))

Q2f)

```
SQL: SELECT Emp.EmployeeName
     FROM Employee AS Emp
       INNER JOIN Employee AS Boss
       ON Emp.BossID = Boss.EmployeeID
     WHERE Emp.EmployeeSalary < (Boss.EmployeeSalary / 2);
```