



DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 8

SQL[DML]

Question

2

- The difference between count(*) and count (attribute name)?
 - ▣ COUNT(*) will count the number of records.
 - ▣ COUNT(column_name) will count the number of records where column_name is not null

Example

3

Customer	ID	Name	Phone	City
	1110	Ahmed	01110034567	Cairo
	1112	Ali	01210034597	Giza
	1113	Mohamed	01515534567	6 October
	1114	Ismael	01220876651	

**Select Count (*)
From Customer**

**Count(*)
4**

**Select Count (ID)
From Customer**

**Count(ID)
4**

**Select Count (City)
From Customer**

**Count(city)
3**

Types of Joins

4

- **Join** — a relational operation that causes two or more tables with a common domain to be combined into a single table or view
 - ▣ **Cross-join**- a join in which there is no joining condition or join condition is always true
 - ▣ **Equi-join** — a join in which the joining condition is based on equality between values in the common columns; common columns **appear redundantly** in the result table
 - ▣ **Natural join** — an equi-join in which one of the duplicate columns is eliminated in the result table
 - ▣ **Outer join** — a join in which rows that do not have matching values in common columns are nonetheless included in the result table (as opposed to *inner join*, in which rows must have matching values in order to appear in the result table)

The common columns in joined tables are usually the primary key of the dominant table and the foreign key of the dependent table in 1:M relationships.

CROSS JOIN

5

Attributes n+m
Cardinality n*m

Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

SELECT * FROM

Student CROSS JOIN

Enrolment

ID	Name	ID	Code
123	John	123	DBS
124	Mary	123	DBS
125	Mark	123	DBS
126	Jane	123	DBS
123	John	124	PRG
124	Mary	124	PRG
125	Mark	124	PRG
126	Jane	124	PRG
123	John	124	DBS
124	Mary	124	DBS

Equi JOIN

6

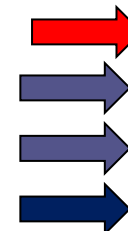
Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

```
SELECT * FROM  
Student, Enrolment  
Where Student.ID=  
Enrolment.ID
```



ID	Name	ID	Code
123	John	123	DBS
124	Mary	124	PRG
124	Mary	124	DBS
126	Jane	126	PRG

NATURAL JOIN

7

Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

SELECT * FROM
Student NATURAL JOIN
Enrolment

ID	Name	Code
123	John	DBS
124	Mary	PRG
124	Mary	DBS
126	Jane	PRG

Join attributes have the same name

Outer Joins

8

- **Left outer join:**
 - ▣ Include the left tuple even if there's no match
- **Right outer join:**
 - ▣ Include the right tuple even if there's no match
- **Full outer join:**
 - ▣ Include the both left and right tuples even if there's no match

Left Join

9

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *  
  from one left join two  
        on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
4	d	.	

Right Join

10

Table Two

X	B
2	x
3	y
5	v

Table One

X	A
1	a
4	d
2	b

```
select *  
  from two right join one  
        on one.x = two.x;
```

X	B	X	A
.		1	a
2	x	2	b
.		4	d

Full Join

11

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *  
  from one full join two  
    on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
.		3	y
4	d	.	
.		5	v

Aliases and 'Self-Joins'

12

Aliases can be used to copy a table, so that it can be combined with itself:

Get the names of all employees who work in the same department as Andy.

Employee

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Aliases and Self-Joins

13

Employee A

A

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Employee B

B

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Aliases and Self-Joins

14

```
SELECT ... FROM Employee A, Employee B ...
```

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	John	Marketing
Mary	Sales	John	Marketing
Peter	Sales	John	Marketing
Andy	Marketing	John	Marketing
Anne	Marketing	John	Marketing
John	Marketing	Mary	Sales
Mary	Sales	Mary	Sales
Peter	Sales	Mary	Sales
Andy	Marketing	Mary	Sales
Anne	Marketing	Mary	Sales

ANSI SQL SELECT

Aliases and Self-Joins

15

```
SELECT ... FROM Employee A, Employee B
WHERE A.Dept = B.Dept
```

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	John	Marketing
Andy	Marketing	John	Marketing
Anne	Marketing	John	Marketing
Mary	Sales	Mary	Sales
Peter	Sales	Mary	Sales
Mary	Sales	Peter	Sales
Peter	Sales	Peter	Sales
John	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing

ANSI SQL SELECT

Aliases and Self-Joins

16

```
SELECT ... FROM Employee A, Employee B
WHERE A.Dept = B.Dept AND B.Name = 'Andy'
```

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing
Anne	Marketing	Andy	Marketing

Aliases and Self-Joins

17

```
SELECT A.Name FROM Employee A, Employee B  
WHERE A.Dept = B.Dept AND B.Name = 'Andy'
```

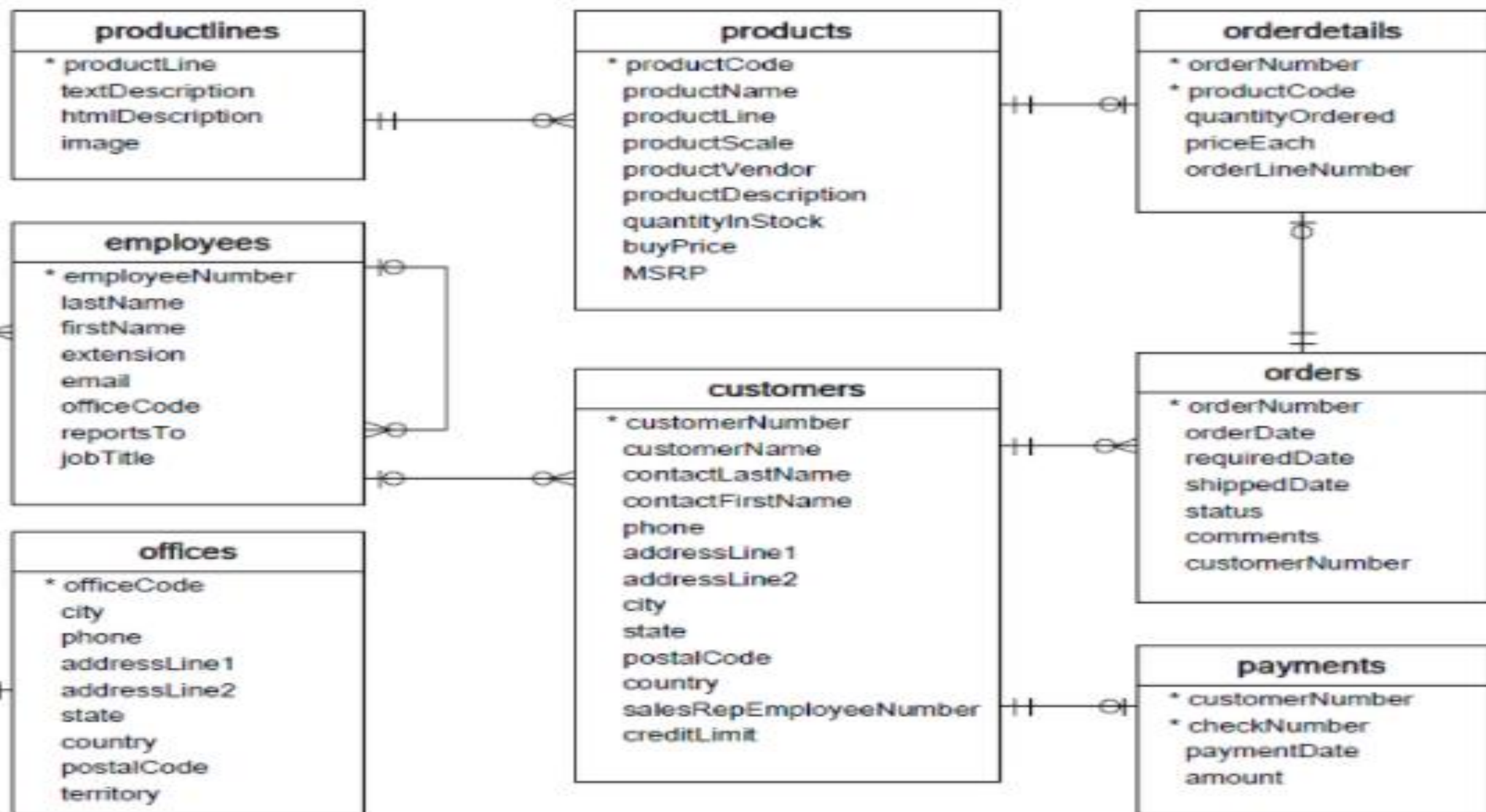
A.Name
John
Andy
Anne

The result is the names of all employees who work in the same department as Andy.

Questions

18

- ❑ **Search and return all order numbers with their products numbers and names.**
- ❑ **Try it as Cross join and Inner join and see the difference.**



Questions

20

- ❑ Search and return all order numbers with their products numbers and names.
- ❑ Try it as Cross join and Inner join and see the difference.

Inner Join: **(Correct Solution)**

```
Select orderdetails.orderNumber,  
products.productCode, products.productName  
From products, orderdetails  
Where products. productCode =  
orderdetails.productCode
```

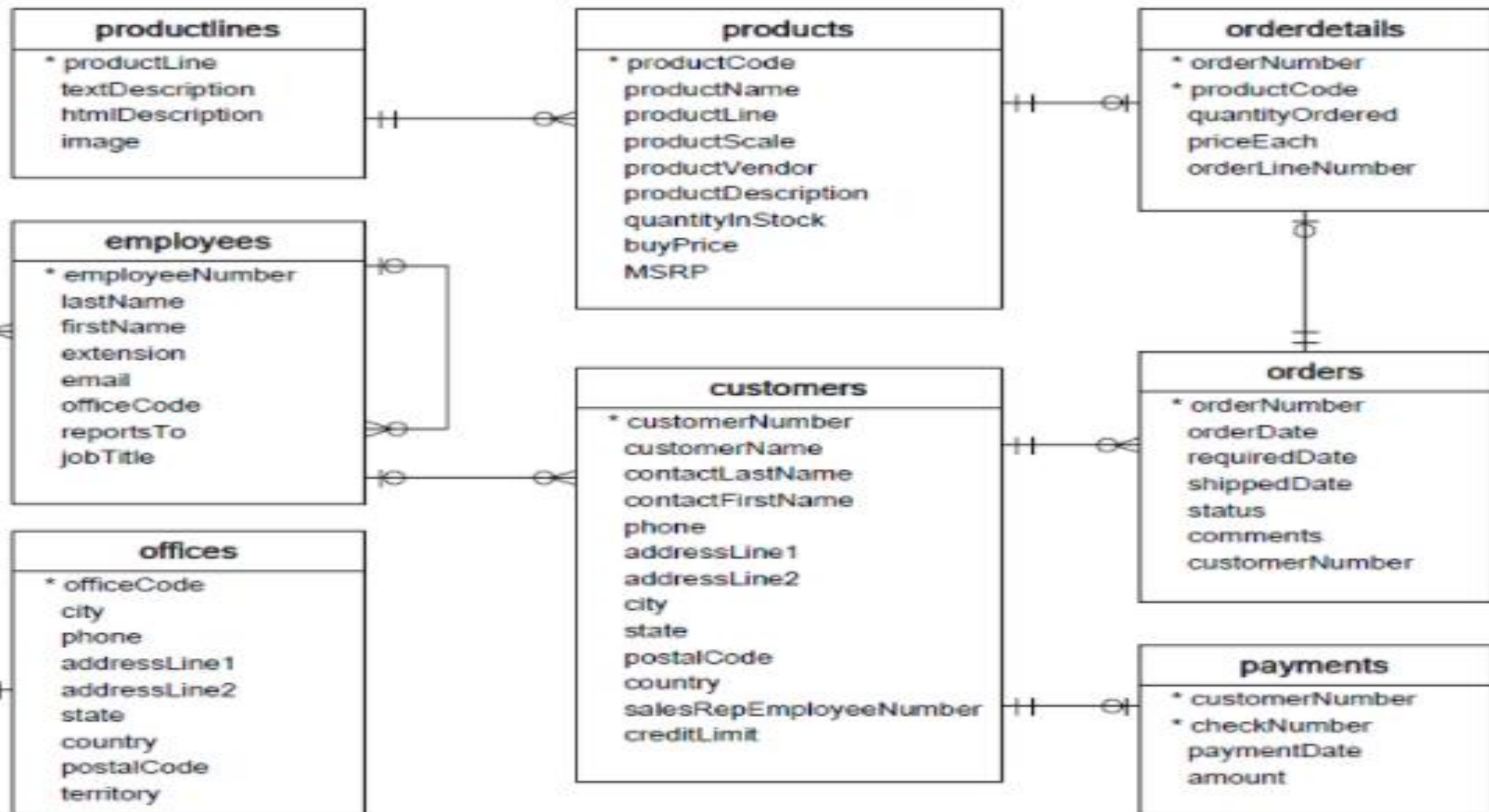
Cross Join:

```
Select orderdetails.orderNumber,  
products.productCode, products.productName  
From products, orderdetails
```

Other Questions for review

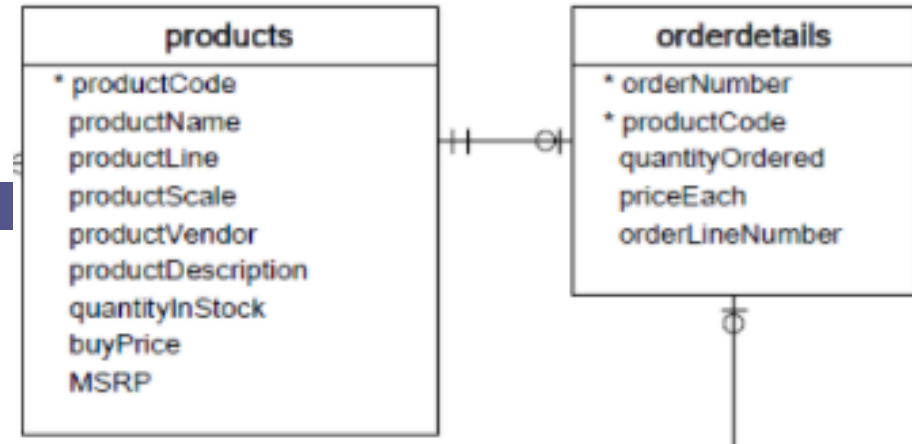
21

1. Get order numbers with their products and the ordered quantities
2. Get employees names working in San Francisco
3. Return all employees names that serve customers in San Francisco
4. Return the names of the customers who paid more than 100,000 in any check
5. Return all customers names and put their amount of payment if they paid more than 100,000. If they didn't pay it, just put their name
6. Return all employees names that worked with customers who paid more than 100,000 in any check.
7. Retrieve each employee with his manager's name.



Review Questions

23



- 1. Get order numbers with their products and the ordered quantities

select orderNumber, orderdetails.quantityOrdered,
products.productName

from orderdetails, products

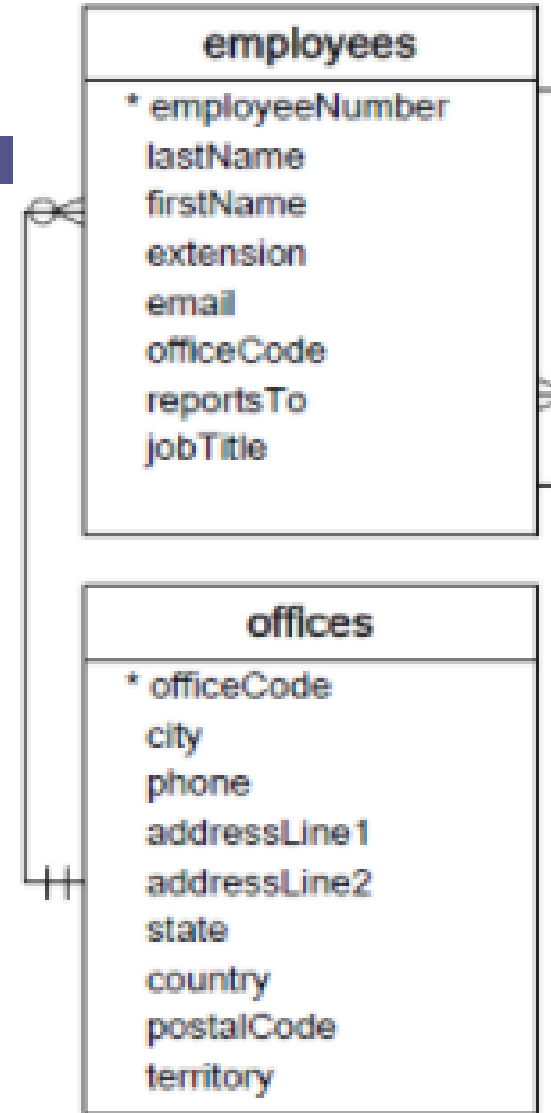
where orderdetails.productCode = products.productCode;

Review Questions

24

□ 2. Get employees names working in San Francisco

```
select firstName, lastName, city
from employees inner join offices
on employees.officeCode = offices.officeCode
where city = "San Francisco";
```



Review Questions

25

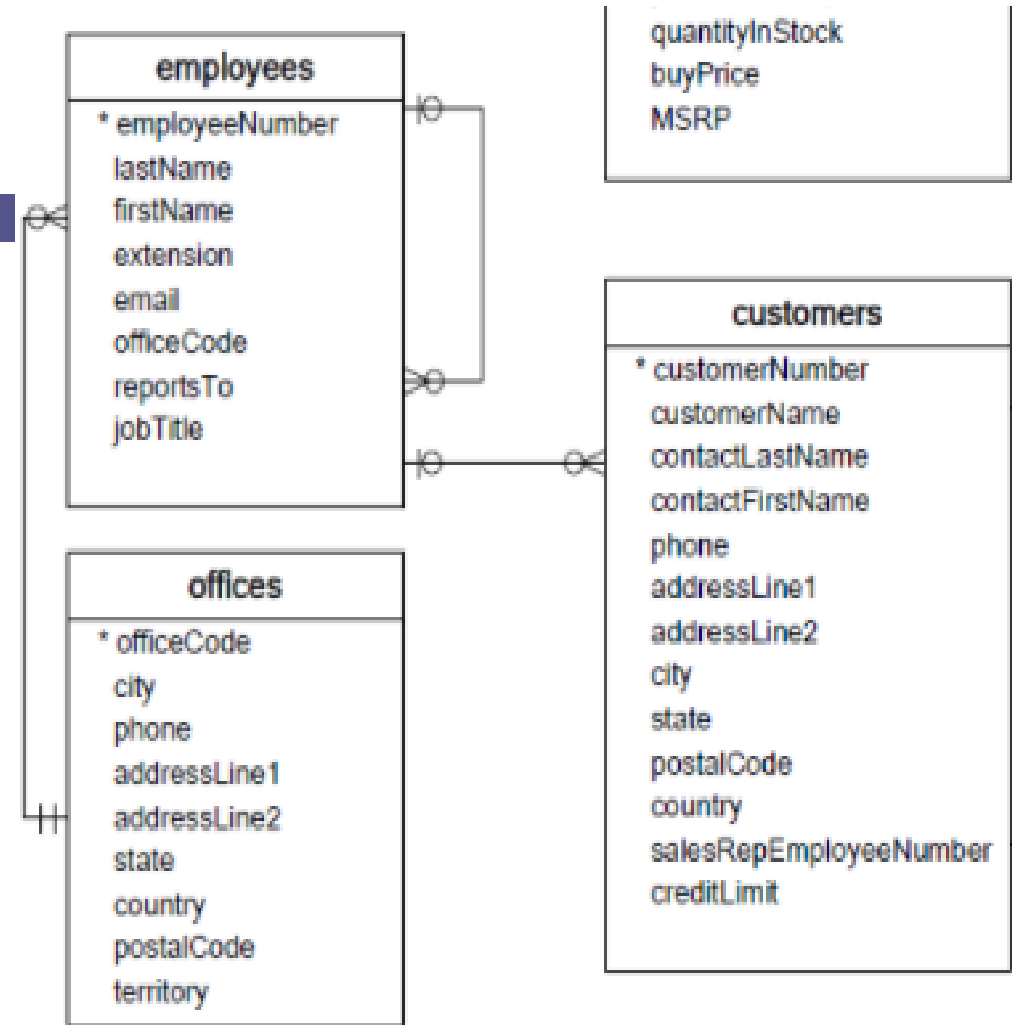
- 3. Return all employees names that serve customers in San Francisco

select employees.firstName, employees.lastName

from employees Inner join customers

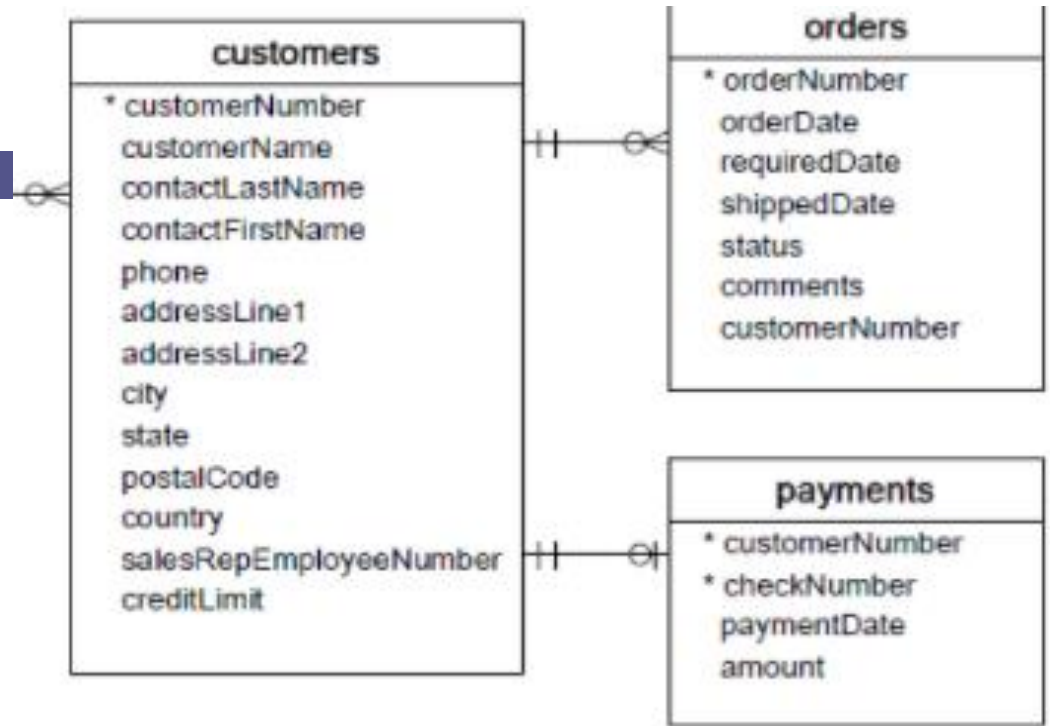
on customers.salesRepEmployeeNumber =
employees.employeeNumber

where customers.city = "San Francisco";



Review Questions

26



- 4. Return the names of the customers who paid more than 100,000 in any check

select customerName, amount

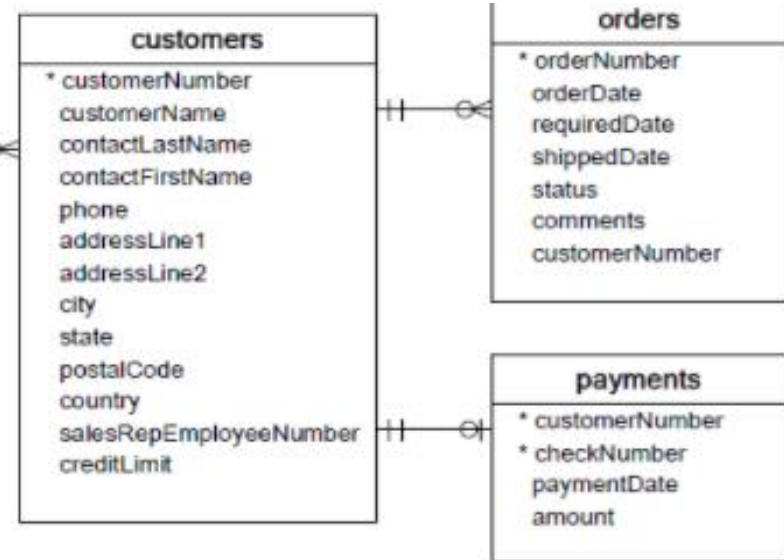
from customers, payments

where customers.customerNumber = payments.customerNumber

and amount>100000;

Review Questions

27



- ❑ 5. Return all customers names and put their amount of payment if they paid more than 100,000. If they didn't pay it, just put their name

select customerName, amount

from customers left join payments

on customers.customerNumber = payments.customerNumber

where amount>100000

Review Questions

28

- 6. Return all employees names that worked with customers who paid more than 100,000 in any check. (Hint: Join 3 tables)

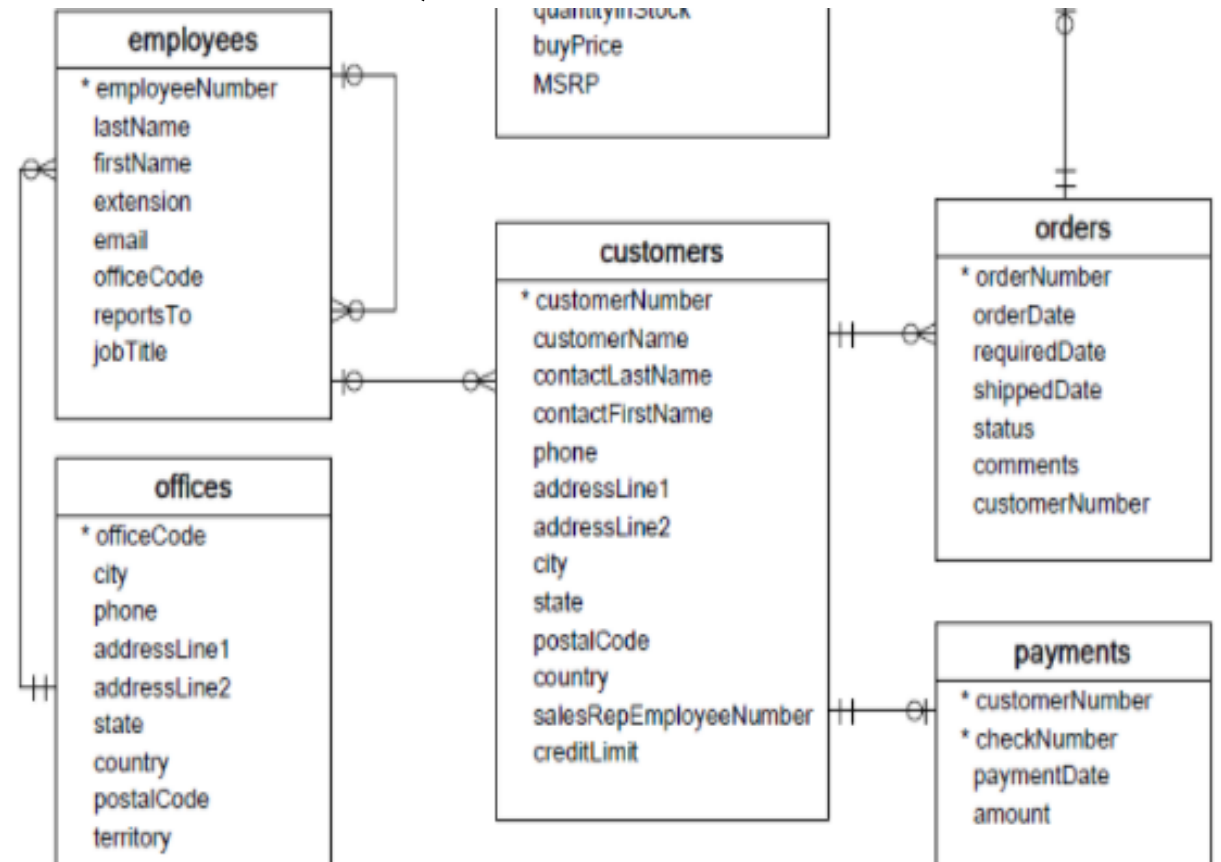
select customerName, amount

from employees, customers, payments

where employees.employeeNumber =
customers.salesRepEmployeeNumber And

customers.customerNumber =
payments.customerNumber

and amount > 100000;



Review Questions

29

- 7. Retrieve each employee name and number with his manager's name. (**Hint: Self Join**)

Select E.employeeNumber, E.firstName, M.firstName
From employees as E, employees as M
Where E.reportsTo = M.employeeNumber;

