

ceu-economics-and-business.github.io

Data modeling. Joins. – ECBS 5146

SQL and Different Shapes of Data

5–6 minutes

Overview

Teaching: 90 min

Questions

- How do you design/model a database?
- In normalized database, the data is structured in a way to avoid data redundancy and support consistency. As analyst, this structure is not always the best fit for analytics. How do you merge one or more tables to get the required data set?

Objectives

- Understanding the basics of data modelling
- Practicing data modelling
- Introducing a larger relational database
- Understanding the difference between different joins
- Present examples and exercise joins

Keywords

#DATA MODELLING

#RELATIONAL DATASETS

#JOINS

Table of Content

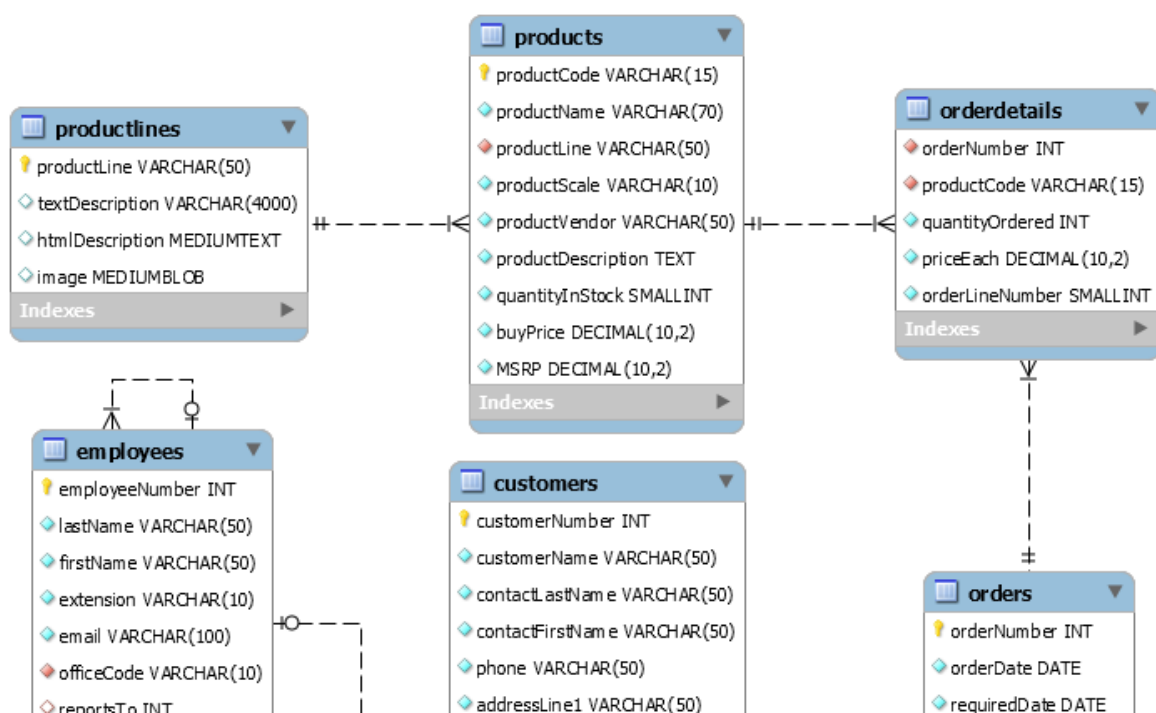
[Lecture PPTX](#)[Data modeling](#)[INNER joins](#)[SELF joins](#)[LEFT joins](#)[Homework](#)

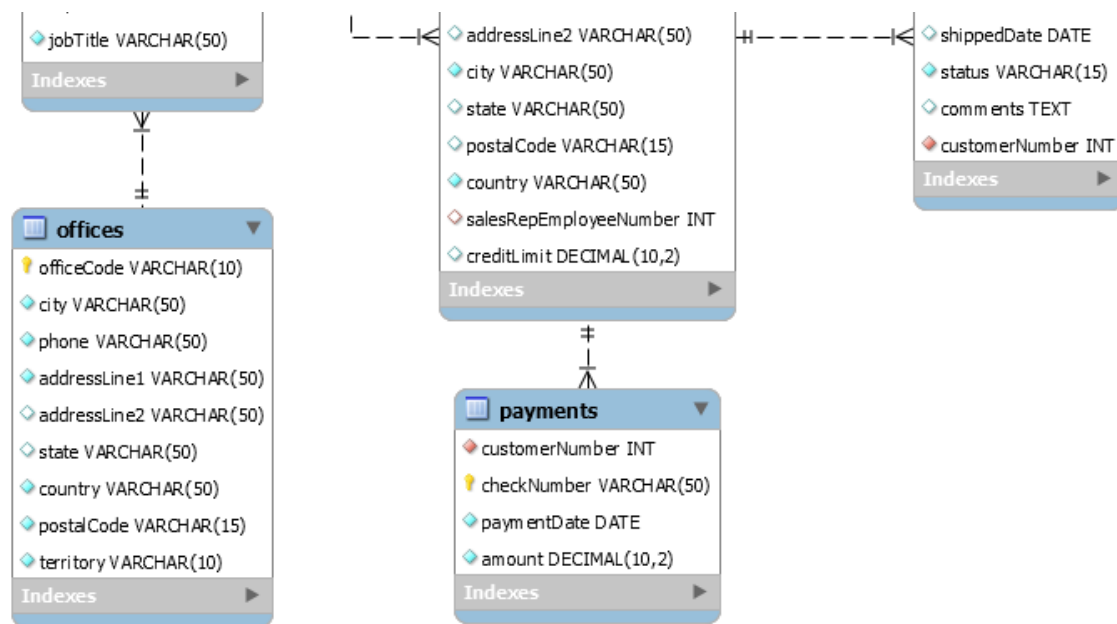
Prerequisites for this chapter

Load [sample database](#) script into your local MySQL instance.

[Credit]

Sample database diagram





Data modeling

There are many types of data models: hierarchical, network, relational etc. Today we will focus on Relational Data modeling.

Several techniques can be used for modeling. Today we will use the most widespread: Entity–relationship modelling (ERM).

Modelling game - University

Let's model in a group of 3 a data structure of a University.

Mandatory elements: Program, Program Coordinator, Course, Prerequisite Course, Student, Instructor.

INNER joins

Syntax

```
SELECT *  
FROM left_table  
INNER JOIN right_table
```

```
ON left_table.id = right_table.id;
```

Basic forms

Join all fields of products and productlines details

```
SELECT *  
FROM products  
INNER JOIN productlines  
ON products.productline = productlines.productline;
```

Same thing with aliasing:

```
SELECT *  
FROM products t1  
INNER JOIN productlines t2  
ON t1.productline = t2.productline;
```

Same thing, but now with USING:

```
SELECT *  
FROM products  
INNER JOIN productlines  
USING(productline);
```

Select specific columns

```
SELECT t1.productName, t1.productLine,  
t2.textDescription  
FROM products t1  
INNER JOIN productlines t2  
ON t1.productline = t2.productline;
```

Exercise1

Join all fields of order and orderdetails

Exercise2

Join all fields of order and orderdetails. Display only orderNumber, status and sum of totalsales (quantityOrdered * priceEach) for each orderNumber.

Multiple INNER joins

```
SELECT *  
FROM left_table  
INNER JOIN right_table  
ON left_table.id = right_table.id  
INNER JOIN another_table  
ON left_table.id = another_table.id;
```

Exercise3

We want to know how the employees are performing. Join orders, customers and employees and return orderDate,lastName, firstName

LEFT JOIN

The next example returns customer info and related orders:

```
SELECT  
    c.customerNumber,  
    customerName,  
    orderNumber,  
    status
```

```
FROM
    customers c
LEFT JOIN orders o
    ON c.customerNumber = o.customerNumber;
```

Difference between LEFT and INNER join

The previous example returns all customers including the customers who have no order. If a customer has no order, the values in the column orderNumber and status are NULL. Try the same query with INNER join.

Difference between LEFT and RIGHT join

Right join is the mirror of the left join, you can achieve the same results with both. Rarely used.

WHERE with joins

```
SELECT
    o.orderNumber,
    customerNumber,
    productCode
FROM
    orders o
INNER JOIN orderDetails
    USING (orderNumber)
WHERE
    orderNumber = 10123;
```

ON

The next query gives the same result yet, the mechanism behind is very different: WHERE does the filtering (`o.orderNumber = 10123`) *after* the join is executed, while in case of ON, the join will happen on the specified subset (`o.orderNumber = 10123`)

```
SELECT
    o.orderNumber,
    customerNumber,
    productCode
FROM
    orders o
INNER JOIN orderDetails d
    ON o.orderNumber = d.orderNumber AND
       o.orderNumber = 10123;
```

SELF JOIN

Employee table represents a hierarchy, which can be flattened with a self join. The next query displays the Manager, Direct report pairs:

```
SELECT
    CONCAT(m.lastName, ', ', m.firstName) AS Manager,
    CONCAT(e.lastName, ', ', e.firstName) AS 'Direct
report'
FROM
    employees e
INNER JOIN employees m ON
    m.employeeNumber = e.reportsTo
ORDER BY
    Manager;
```

Question

Why President is not in the list?

Homework 4

- INNER join orders,orderdetails,products and customers. Return back:
- orderNumber
- priceEach
- quantityOrdered
- productName
- productLine
- city
- country
- orderDate
- Upload your solution to your GitHub repo in a folder called HW4
- Submit GitHub repo link to moodle when you are ready