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;

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;