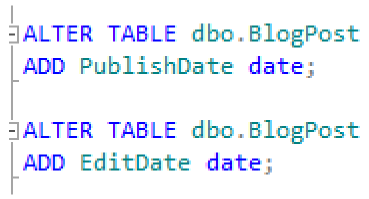
BUSINESS ACADEMY
AARHUS

Databases & XML (3) – 22.02.2016

Time	Activity
8.30	Introduction and code examples
8.40	ER exercise
9.40	Solutions from students
10.00	Break
10.30	Functions and stored procedures
10.50	Continue SQL assignments
11.55	Homework and preparation

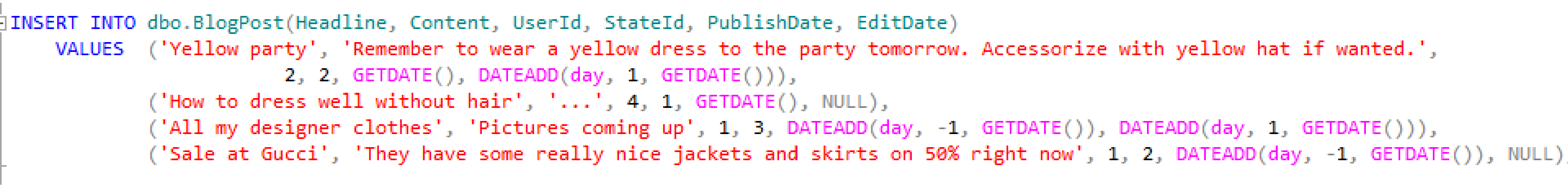
Today’s agenda

BUSINESS ACADEMY AARHUS

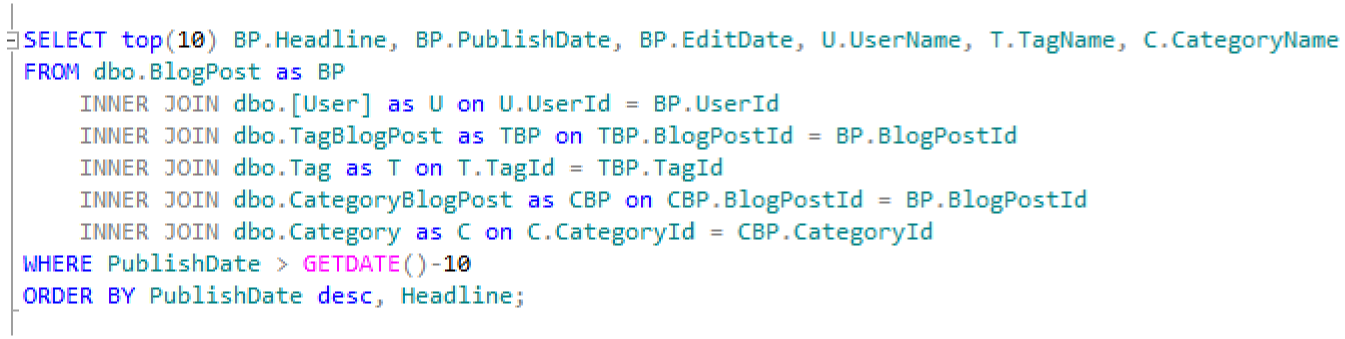


```sql
ALTER TABLE dbo.BlogPost
ADD PublishDate date;

ALTER TABLE dbo.BlogPost
ADD EditDate date;
```



```sql
INSERT INTO dbo.BlogPost(Headline, Content, UserId, StateId, PublishDate, EditDate)
    VALUES  ('Yellow party', 'Remember to wear a yellow dress to the party tomorrow. Accessorize with yellow hat if wanted.',
                    2, 2, GETDATE(), DATEADD(day, 1, GETDATE())),
            ('How to dress well without hair', '...', 4, 1, GETDATE(), NULL),
            ('All my designer clothes', 'Pictures coming up', 1, 3, DATEADD(day, -1, GETDATE()), DATEADD(day, 1, GETDATE())),
            ('Sale at Gucci', 'They have some really nice jackets and skirts on 50% right now', 1, 2, DATEADD(day, -1, GETDATE()), NULL)
```



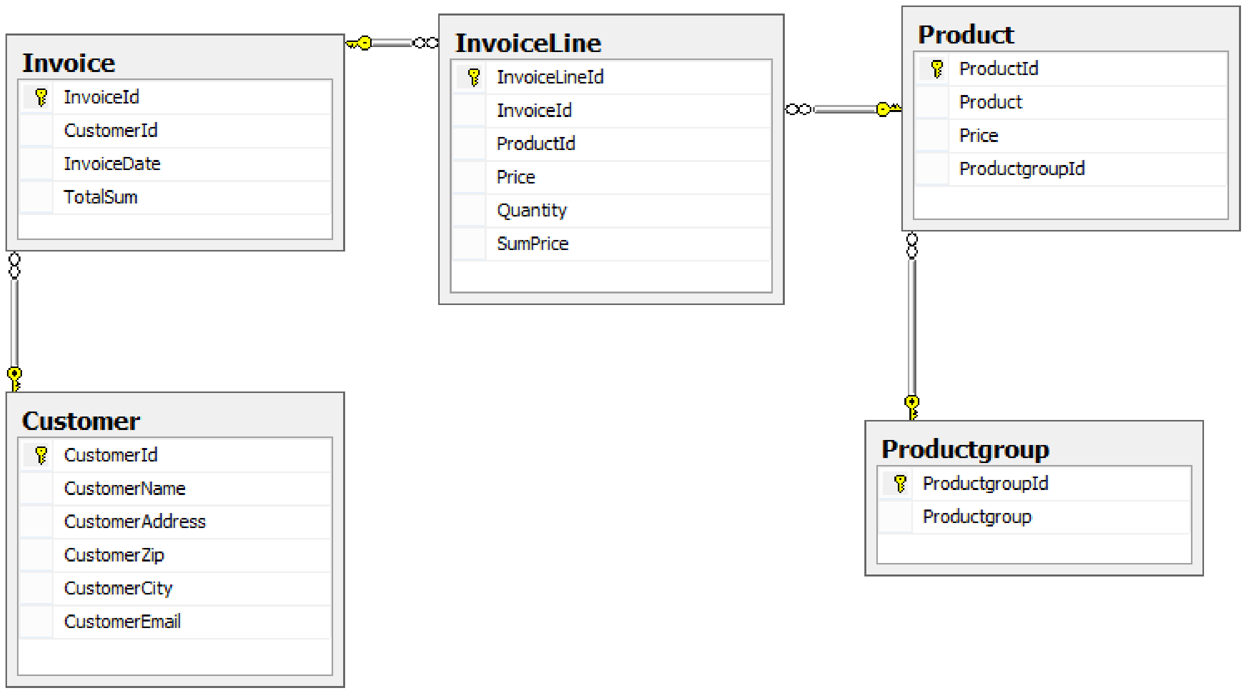
```sql
SELECT top(10) BP.Headline, BP.PublishDate, BP.EditDate, U.UserName, T.TagName, C.CategoryName
FROM dbo.BlogPost as BP
    INNER JOIN dbo.[User] as U on U.UserId = BP.UserId
    INNER JOIN dbo.TagBlogPost as TBP on TBP.BlogPostId = BP.BlogPostId
    INNER JOIN dbo.Tag as T on T.TagId = TBP.TagId
    INNER JOIN dbo.CategoryBlogPost as CBP on CBP.BlogPostId = BP.BlogPostId
    INNER JOIN dbo.Category as C on C.CategoryId = CBP.CategoryId
WHERE PublishDate > GETDATE()-10
ORDER BY PublishDate desc, Headline;
```

SQL 2-2

BUSINESS ACADEMY
AARHUS

ER exercise

BUSINESS ACADEMY AARHUS



ER exercise

BUSINESS ACADEMY AARHUS

Break

BUSINESS ACADEMY AARHUS

- The purpose of a user-defined function (UDF) is to encapsulate logic that calculates something, possibly based on input parameters, and return a result.

- Scalar UDFs return a single value
- Table-valued UDFs return a table

- You can use UDFs in queries
  - Scalar UDFs anywhere in the query where an expression that returns a single value can appear
  - Table-valued UDFs in the FROM clause

- UDFs are not allowed to have any side effects (some build-in functions are banned because of this)
- Must include a RETURN clause

Functions

BUSINESS ACADEMY AARHUS

- COUNT
- SUM
- AVG
- MIN
- MAX

- These functions operate on a single column of a table and return a single value
- SUM and AVG may be used on numeric fields only
- Each function eliminates nulls first and operates only on the remaining non-null values
- The exception: COUNT(*) that counts all the rows of a table, regardless of whether nulls or duplicate values occur
- Can only be used in the SELECT list and in the HAVING clause
- If the SELECT contains other values than aggregate functions, a GROUP BY clause must be used

Aggregate functions

GROUP BY

- Aggregate functions is often used to get the totals at the bottom of a report, since they condense all the detailed data in the report into a single summary row of data
- However, it is often useful to have subtotals in reports – which we can use the GROUP BY clause to
- A query that includes the GROUP BY clause is called a **grouped query**, because it groups the data from the SELECT table(s) and produces a single summary row for each group
- When GROUP BY is used, each item in the SELECT list must be single-valued per group
- All the column names in the SELECT list must appear in the GROUP BY clause unless the name is used only in an aggregate function
- There may be column names in the GROUP BY clause that do not appear in the SELECT list
- Order: WHERE – GROUP BY – ORDER BY

- Stored procedures are server-side routines that encapsulate T-SQL code
- Can have input and output parameters
- Can return result sets of queries
- Are allowed to invoke code that has side effects
- Better control of security – you can validate requests
- Supports error handling
- Better performance by reuse of previously cached execution plans and by reduction of network traffic

Stored procedures

Break

BUSINESS ACADEMY AARHUS

- Lesson 01
  - SQL-1 poll app
  - SQL-2 Start of the blog
- Lesson 02
  - SQL-3 Date and recursive categories
- Lesson 03
  - Lesson 03 exercise – web shop
  - SQL-4 Automation

Continue SQL assignment

Next week's topic: **Transactions and Triggers**

Read:
T-SQL chapter 9
Beginning C# 2008 Databases – Transactions (on Fronter)

Exercises (in this order):
SQL-1 poll app
SQL-2 Start of the blog
SQL-3 Date and recursive categories
SQL-4 Automation

Homework and preparation

BUSINESS ACADEMY AARHUS