Datenbanken und SQL

(Woche 2 - Tag 5)

Agenda

Abfragen (über 1 Tabelle)

- Definition + Motivation
- Syntax
  - Alle Attribute aller Datensätze
  - Ausgewählte Attribute aller Datensätze
  - Bedingte Auswahl der Datensätze
  - Vermeidung von Ausgabe-Dubletten
  - Sortierte Ausgabe
    - 1 Sortierkategorie
    - mehrere Sortierkategorien
    - Sortierrichtung
  - Quantitative Begrenzung der Ausgabe
    - vom ersten Datensatz an
    - nach einer Anzahl von übersprungenen Datensätzen

Abfragen

Definition

- Abfragen können gezielt Informationen aus einer Datenbank **„selektieren“**.
- Der entsprechende SQL-Befehl lautet daher auch **„SELECT“**.
- Hierfür werden wir mitteilen müssen, welche **Attributwerte** wir ausgeben lassen möchten und (aus Performance-Gründen) in welchen **Tabellen** sich diese befinden. Optional können wir mittels **Bedingungen** (WHERE-Klauseln) die zu berücksichtigenden Datensätze filtern.
- Wie schon beim „DELETE“- und „UPDATE“-Befehl wird auch beim Aufruf eines SELECT-Statements eine **Schleife** gestartet, die alle Datensätze der angegebenen Tabellen durchlaufen wird. Pro Datensatz wird die gegebenenfalls vorliegende Bedingung überprüft. Die Ausgabe geschieht anschließend pro Datensatz, sofern die Bedingung erfüllt ist.
- Falls **keine explizite Bedingung** benannt wurde, erfolgt die Ausgabe für **jeden Datensatz**.
- Der SELECT-Befehl wird üblicherweise zur **DML** gezählt. Die Zuordnung zur DQL (Data Querry Language) ist hingegen kaum gebräuchlich.
- Die **Ausgabe** geschieht in Form einer **Tabelle**.

Motivation

- Nach fast 2 Wochen, in denen wir lernten, Datenbanken zu modellieren, zu implementieren, mit Daten zu füllen und gegebenenfalls Struktur und Datenbestand zu korrigieren, werden wir uns nun erstmalig damit befassen, Datenbanken auch tatsächlich zu **nutzen**.
- Die **Motivation** von Abfragen erklärt sich mithin von selbst.
- Wir werden uns im Folgenden zunächst mit einfachen Abfragen beschäftigen, deren Ausgabe sich kaum von dem unterscheidet, was wir zuvor mittels INSERT in die Datenbank einpflegten, werden aber im Laufe der nächsten Tage lernen, deutlich **„intelligentere“ Abfragen** zu formulieren.

Syntax

Alle Attribute aller Datensätze – SELECT * FROM Tabellenname;

Beispiel: Ausgabe aller Attribute aller Datensätze aus Kunde:

```
SELECT * FROM Kunde;
```

Kunde_ID	Vorname	Nachname	Email
1	Elli	Rot	rot@xyz.de
2	Vera	Deise	deise@xyz.de
3	Witali	Myrnow	myr@xyz.de
4	Rita	Myrnow	myr@xyz.de
5	Eva	Hahn	ehahn@xyz.de
6	Gala	Nieda	gala@xyz.de
7	Peter	Kaufnix	nix@xyz.de

Ausgewählte Attribute aller Datensätze – **SELECT** Attribut1, Attribut2, … **FROM** Tabellenname;

Beispiel: Für alle Kunden sollen Vor- und Nachname ausgegeben werden:

SELECT Vorname, Nachname **FROM** Kunde;

Vorname	Nachname
Elli	Rot
Vera	Deise
Witali	Myrnow
Rita	Myrnow
Eva	Hahn
Gala	Nieda
Peter	Kaufnix

Bedingte Auswahl der Datensätze – **SELECT** Attribut1**,** Attribut2**,** … **FROM** Tabellenname **WHERE** …**;**

Beispiel: Kunde-ID und Email-Adresse aller Kunden mit ID > 3.

SELECT Kunde_ID**,** Email **FROM** Kunde **WHERE** Kunde_ID **>** 3**;**

kunde_id	email
4	myr@xyz.de
5	ehahn@xyz.de
6	gala@xyz.de
7	nix@xyz.de

Ergänzung:

Falls für alle Kunden mit ID>3 **alle Attributwerte** ausgegeben werden sollen,
so kann erneut die Kurzschreibweise genutzt werden:

SELECT * FROM Kunde **WHERE** Kunde_ID**>**3**;**

Die Funktionalität zeigt sich an den 3 folgenden Beispielen:

SELECT Nachname, Email **FROM** Kunde WHERE Kunde_ID>1 AND Kunde_ID<6;

(zur Erinnerung: Unsere Datenbank besitzt zwei Kunden mit Nachnamen „Myrnow“)

Nachname	Email
Deise	deise@xyz.de
Myrnow	myr@xyz.de
Myrnow	myr@xyz.de
Hahn	ehahn@xyz.de

SELECT DISTINCT Nachname, Email **FROM** Kunde WHERE Kunde_ID>1 AND Kunde_ID<6;

Nachname	Email
Deise	deise@xyz.de
Myrnow	myr@xyz.de
Hahn	ehahn@xyz.de

SELECT DISTINCT Vorname, Nachname, Email **FROM** Kunde WHERE Kunde_ID>1 AND Kunde_ID<6;

Vorname	Nachname	Email
Vera	Deise	deise@xyz.de
Witali	Myrnow	myr@xyz.de
Rita	Myrnow	myr@xyz.de
Eva	Hahn	ehahn@xyz.de

Sortierte Ausgabe

1 Sortierkriterium – SELECT … FROM … ORDER BY Attribut;

Die Funktionalität zeigt sich an den folgenden Beispielen:

SELECT * FROM Kunde **ORDER BY** Nachname**;**

Kunde_ID	Vorname	Nachname ▲ 1	Email
2	Vera	Deise	deise@xyz.de
5	Eva	Hahn	ehahn@xyz.de
7	Peter	Kaufnix	nix@xyz.de
3	Witali	Myrnow	myr@xyz.de
4	Rita	Myrnow	myr@xyz.de
6	Gala	Nieda	gala@xyz.de
1	Elli	Rot	rot@xyz.de

SELECT * FROM Abrechnung **ORDER BY** Datum**;**

Abrechnung_ID	Kunde_id	Datum ▲ 1
1	1	2021-05-05
2	3	2021-10-07
3	2	2021-10-11
4	3	2021-10-16
5	5	2021-10-25
6	4	2021-11-03
7	3	2021-11-05
8	2	2021-11-09
9	1	2021-11-17
10	7	2022-02-14

Mehrere Sortierkriterien – SELECT … FROM … ORDER BY Attribut, … ;

Die Funktionalität eines weiteren Sortierkriteriums zeigt sich an folgendem Beispiel:

SELECT * FROM Kunde **ORDER BY** Nachname**;**

Kunde_ID	Vorname	Nachname ▲ 1	Email
2	Vera	Deise	deise@xyz.de
5	Eva	Hahn	ehahn@xyz.de
7	Peter	Kaufnix	nix@xyz.de
3	Witali	Myrnow	myr@xyz.de
4	Rita	Myrnow	myr@xyz.de
6	Gala	Nieda	gala@xyz.de
1	Elli	Rot	rot@xyz.de

SELECT * FROM Kunde **ORDER BY** Nachname, Vorname**;**

Kunde_ID	Vorname ▲ 2	Nachname ▲ 1	Email
2	Vera	Deise	deise@xyz.de
5	Eva	Hahn	ehahn@xyz.de
7	Peter	Kaufnix	nix@xyz.de
4	Rita	Myrnow	myr@xyz.de
3	Witali	Myrnow	myr@xyz.de
6	Gala	Nieda	gala@xyz.de
1	Elli	Rot	rot@xyz.de

Sortierrichtung – SELECT … FROM … ORDER BY Attribut [ASC oder DESC], … ;

Der Unterschied von ASC (ascendent) und DESC (descendent) zeigt sich an folgendem Beispiel:

SELECT * FROM Produkt **ORDER BY** Euro_Preis **ASC;**

Ascendent („aufsteigend“) ist die **Default-Einstellung**
und kann daher ungenannt bleiben.

Produkt_id	Hersteller_ID	Produkt_Name	Euro_Preis
1	2	tool 2.0	15.98
2	2	tool 3.1	22.75
3	1	solver 1000	31.69
4	1	solver premium	45.05
5	3	Do IT edition 1	98.00
6	5	TroppoCaro	1000.00

SELECT * FROM Produkt **ORDER BY** Euro_Preis **DESC;**

Descendent („absteigend“) ist hingegen **obligatorisch**,
sofern man die Reihenfolge umdrehen möchte.

Produkt_id	Hersteller_ID	Produkt_Name	Euro_Preis
6	5	TroppoCaro	1000.00
5	3	Do IT edition 1	98.00
4	1	solver premium	45.05
3	1	solver 1000	31.69
2	2	tool 3.1	22.75
1	2	tool 2.0	15.98

Begrenzte Ausgabe – … ORDER BY … LIMIT …;

Wir zeigen die beiden Versionen an folgenden Beispielen:

Ausgabe von Beginn an, Beispiel: Name und Preis der **3** teuersten Produkte:

```
SELECT Produkt_Name, Euro_Preis  FROM Produkt ORDER BY Euro_Preis DESC LIMIT 3;
```

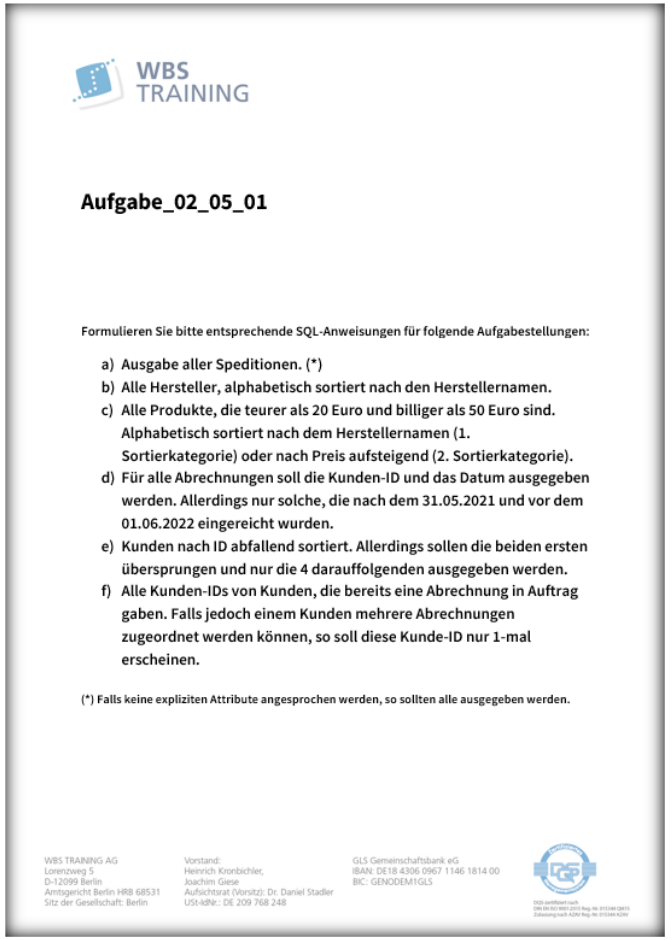
Produkt_Name	Euro_Preis
TroppoCaro	1000.00
Do IT edition 1	98.00
solver premium	45.05

Ausgabe nach einer Anzahl von Übersprungenen, Beispiel: Name und Preis der **2** teuersten Produkte NACH **1** übersprungenen Produkt:

```
SELECT Produkt_Name, Euro_Preis  FROM Produkt ORDER BY Euro_Preis DESC LIMIT 1,2;
```

Produkt_Name	Euro_Preis
Do IT edition 1	98.00
solver premium	45.05

Gemeinsame Übung („Live-Coding“) -> A_02_05_01



WBS TRAINING

Aufgabe_02_05_01

Formulieren Sie bitte entsprechende SQL-Anweisungen für folgende Aufgabestellungen:

a) Ausgabe aller Speditionen. (*)
b) Alle Hersteller, alphabetisch sortiert nach den Herstellernamen.
c) Alle Produkte, die teurer als 20 Euro und billiger als 50 Euro sind. Alphabetisch sortiert nach dem Herstellernamen (1. Sortierkategorie) oder nach Preis aufsteigend (2. Sortierkategorie).
d) Für alle Abrechnungen soll die Kunden-ID und das Datum ausgegeben werden. Allerdings nur solche, die nach dem 31.05.2021 und vor dem 01.06.2022 eingereicht wurden.
e) Kunden nach ID abfallend sortiert. Allerdings sollen die beiden ersten übersprungen und nur die 4 darauffolgenden ausgegeben werden.
f) Alle Kunden-IDs von Kunden, die bereits eine Abrechnung in Auftrag gaben. Falls jedoch einem Kunden mehrere Abrechnungen zugeordnet werden können, so soll diese Kunde-ID nur 1-mal erscheinen.

(*) Falls keine expliziten Attribute angesprochen werden, so sollten alle ausgegeben werden.

WBS TRAINING AG
Lorenzweg 5
D-12099 Berlin
Amtsgericht Berlin HRB 68531
Sitz der Gesellschaft: Berlin

Vorstand:
Heinrich Kronbichler,
Joachim Giese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IdNr.: DE 209 768 248

GLS Gemeinschaftsbank eG
IBAN: DE18 4306 0967 1146 1814 00
BIC: GENODEM1GLS

Vielen Dank für Ihre Aufmerksamkeit!