

Praktische SQL-Befehle 2

Datenbanksysteme I
WiSe 2018/2019

Todor Ivanov

Praktische SQL-Befehle

- Functions
- Views
- Triggers

→ Voraussetzung: Laptop + MySQL/ MariaDB + Workbench!

Airport Database: 7 Tables (MySQL Workbench View)

Table: flightexecution Columns: <u>FlightNo</u> varchar(7) PK <u>DepartureDateAndTimeUTC</u> timestamp PK ICAO_Code_Origin varchar(4) ICAO_Code_Destination varchar(4) PlaneID int(11) FlightDurationInMinutes int(11)		<table border="1"> <thead> <tr> <th>FlightNo</th> <th>DepartureDateAndTimeUTC</th> <th>ICAO_Code_Origin</th> <th>ICAO_Code_Destination</th> <th>PlaneID</th> <th>FlightDurationInMinutes</th> </tr> </thead> <tbody> <tr> <td>IBE1684</td> <td>2018-12-31 10:30:45</td> <td>LIRA</td> <td>ENGM</td> <td>2</td> <td>200</td> </tr> <tr> <td>IBE1764</td> <td>2018-12-31 10:30:45</td> <td>EGLC</td> <td>EDDT</td> <td>1</td> <td>180</td> </tr> <tr> <td>IBE1846</td> <td>2018-12-31 10:30:45</td> <td>EDDF</td> <td>EDDL</td> <td>3</td> <td>160</td> </tr> </tbody> </table>	FlightNo	DepartureDateAndTimeUTC	ICAO_Code_Origin	ICAO_Code_Destination	PlaneID	FlightDurationInMinutes	IBE1684	2018-12-31 10:30:45	LIRA	ENGM	2	200	IBE1764	2018-12-31 10:30:45	EGLC	EDDT	1	180	IBE1846	2018-12-31 10:30:45	EDDF	EDDL	3	160
FlightNo	DepartureDateAndTimeUTC	ICAO_Code_Origin	ICAO_Code_Destination	PlaneID	FlightDurationInMinutes																					
IBE1684	2018-12-31 10:30:45	LIRA	ENGM	2	200																					
IBE1764	2018-12-31 10:30:45	EGLC	EDDT	1	180																					
IBE1846	2018-12-31 10:30:45	EDDF	EDDL	3	160																					
Table: address Columns: <u>AddressID</u> int(11) PK ZIPCode varchar(10) Town varchar(100) Street varchar(100) CountryISO3166_2LetterCode char(2)		<table border="1"> <thead> <tr> <th>AddressID</th> <th>ZIPCode</th> <th>Town</th> <th>Street</th> <th>CountryISO3166_2LetterCode</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>13405</td> <td>Berlin</td> <td>Flughafen Tegel</td> <td>DE</td> </tr> <tr> <td>2</td> <td>40474</td> <td>Düsseldorf</td> <td>Flughafenstraße 120</td> <td>DE</td> </tr> <tr> <td>3</td> <td>60547</td> <td>Frankfurt am Main</td> <td>Frankfurt Flughafen</td> <td>DE</td> </tr> </tbody> </table>	AddressID	ZIPCode	Town	Street	CountryISO3166_2LetterCode	1	13405	Berlin	Flughafen Tegel	DE	2	40474	Düsseldorf	Flughafenstraße 120	DE	3	60547	Frankfurt am Main	Frankfurt Flughafen	DE				
AddressID	ZIPCode	Town	Street	CountryISO3166_2LetterCode																						
1	13405	Berlin	Flughafen Tegel	DE																						
2	40474	Düsseldorf	Flughafenstraße 120	DE																						
3	60547	Frankfurt am Main	Frankfurt Flughafen	DE																						
Table: reservation Columns: <u>ReservationID</u> int(11) PK <u>CustomerID</u> int(11) NoReservedSeats int(11) Comment varchar(1000) FlightNo varchar(7) DepartureDateAndTimeUTC timestamp		<table border="1"> <thead> <tr> <th>ReservationID</th> <th>CustomerID</th> <th>NoReservedSeats</th> <th>Comment</th> <th>FlightNo</th> <th>DepartureDateAndTimeUTC</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>16579435</td> <td>1</td> <td></td> <td>LH1973</td> <td>2018-12-31 10:30:45</td> </tr> <tr> <td>2</td> <td>35943147</td> <td>1</td> <td></td> <td>LH3584</td> <td>2018-12-31 10:30:45</td> </tr> <tr> <td>3</td> <td>15943155</td> <td>2</td> <td>Die Plätze sollten nebeneinander liegen</td> <td>LH1167</td> <td>2018-12-31 10:30:45</td> </tr> </tbody> </table>	ReservationID	CustomerID	NoReservedSeats	Comment	FlightNo	DepartureDateAndTimeUTC	1	16579435	1		LH1973	2018-12-31 10:30:45	2	35943147	1		LH3584	2018-12-31 10:30:45	3	15943155	2	Die Plätze sollten nebeneinander liegen	LH1167	2018-12-31 10:30:45
ReservationID	CustomerID	NoReservedSeats	Comment	FlightNo	DepartureDateAndTimeUTC																					
1	16579435	1		LH1973	2018-12-31 10:30:45																					
2	35943147	1		LH3584	2018-12-31 10:30:45																					
3	15943155	2	Die Plätze sollten nebeneinander liegen	LH1167	2018-12-31 10:30:45																					
Table: airport Columns: <u>ICAO_Code</u> varchar(4) PK AirportName varchar(100) AddressID int(11)		<table border="1"> <thead> <tr> <th>ICAO_Code</th> <th>AirportName</th> <th>AddressID</th> </tr> </thead> <tbody> <tr> <td>CYYZ</td> <td>Toronto Pearson International Airport</td> <td>10</td> </tr> <tr> <td>EDDF</td> <td>Frankfurt am Main</td> <td>3</td> </tr> <tr> <td>EDDL</td> <td>Düsseldorf International</td> <td>2</td> </tr> </tbody> </table>	ICAO_Code	AirportName	AddressID	CYYZ	Toronto Pearson International Airport	10	EDDF	Frankfurt am Main	3	EDDL	Düsseldorf International	2												
ICAO_Code	AirportName	AddressID																								
CYYZ	Toronto Pearson International Airport	10																								
EDDF	Frankfurt am Main	3																								
EDDL	Düsseldorf International	2																								
Table: country Columns: <u>CountryISO3166_2LetterCode</u> char(2) PK CountryName varchar(100)		<table border="1"> <thead> <tr> <th>CountryISO3166_2LetterCode</th> <th>CountryName</th> </tr> </thead> <tbody> <tr> <td>CA</td> <td>Canada</td> </tr> <tr> <td>DE</td> <td>Germany</td> </tr> <tr> <td>FR</td> <td>France</td> </tr> </tbody> </table>	CountryISO3166_2LetterCode	CountryName	CA	Canada	DE	Germany	FR	France																
CountryISO3166_2LetterCode	CountryName																									
CA	Canada																									
DE	Germany																									
FR	France																									
Table: customer Columns: <u>ID</u> int(11) PK LastName varchar(50) FirstName varchar(50) AddressID int(11)		<table border="1"> <thead> <tr> <th>ID</th> <th>LastName</th> <th>FirstName</th> <th>AddressID</th> </tr> </thead> <tbody> <tr> <td>15943155</td> <td>Müller</td> <td>Elias</td> <td>13</td> </tr> <tr> <td>15946782</td> <td>Richter</td> <td>Thomas</td> <td>20</td> </tr> <tr> <td>16579435</td> <td>Schmidt</td> <td>Alexander</td> <td>11</td> </tr> </tbody> </table>	ID	LastName	FirstName	AddressID	15943155	Müller	Elias	13	15946782	Richter	Thomas	20	16579435	Schmidt	Alexander	11								
ID	LastName	FirstName	AddressID																							
15943155	Müller	Elias	13																							
15946782	Richter	Thomas	20																							
16579435	Schmidt	Alexander	11																							
Table: plane Columns: <u>PlaneID</u> int(11) PK NoOfSeats int(11) PlaneType varchar(100) PlaneName varchar(100)		<table border="1"> <thead> <tr> <th>PlaneID</th> <th>NoOfSeats</th> <th>PlaneType</th> <th>PlaneName</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>555</td> <td>Airbus</td> <td>Airbus A380</td> </tr> <tr> <td>2</td> <td>85</td> <td>Fokker</td> <td>Fokker F100</td> </tr> <tr> <td>3</td> <td>162</td> <td>Boeing</td> <td>Boeing 737</td> </tr> </tbody> </table>	PlaneID	NoOfSeats	PlaneType	PlaneName	1	555	Airbus	Airbus A380	2	85	Fokker	Fokker F100	3	162	Boeing	Boeing 737								
PlaneID	NoOfSeats	PlaneType	PlaneName																							
1	555	Airbus	Airbus A380																							
2	85	Fokker	Fokker F100																							
3	162	Boeing	Boeing 737																							

Aggregatfunktionen in SQL

- Eine Besonderheit von SQL sind die Operatoren, die Berechnungen über Gruppen von Tupeln anstellen.
- Die sog. Aggregatfunktionen können in der **select**-Klausel anstelle von einzelnen Attributen angegeben werden.

Beispiele für Funktionen:

- **min(A)** zur Berechnung des Minimalwerts aller Tupel unter dem Attribut A.
- **max(A)** zur Berechnung des Maximalwerts aller Tupel unter dem Attribut A.
- **avg([distinct] A)** zur Berechnung des Durchschnittswerts aller Tupel unter dem Attribut A, wobei unter Angabe von **distinct** mehrfach gleiche Werte nur einmal in die Berechnung eingehen.

Function Syntax (<https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html>)

```
1 CREATE
2   [DEFINER = { user | CURRENT_USER }]
3   PROCEDURE sp_name ([proc_parameter[,...]])
4   [characteristic ...] routine_body
5
6 CREATE
7   [DEFINER = { user | CURRENT_USER }]
8   FUNCTION sp_name ([func_parameter[,...]])
9   RETURNS type
10  [characteristic ...] routine_body
11
12 proc_parameter:
13   [ IN | OUT | INOUT ] param_name type
14
15 func_parameter:
16   param_name type
17
18 type:
19   Any valid MySQL data type
20
21 characteristic:
22   COMMENT 'string'
23   | LANGUAGE SQL
24   | [NOT] DETERMINISTIC
25   | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
26   | SQL SECURITY { DEFINER | INVOKER }
27
28 routine_body:
29   Valid SQL routine statement
```

Beispiel: Hello funktion

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `hello`(s CHAR(20)) RETURNS char(50) CHARSET latin1
2   DETERMINISTIC
3   BEGIN
4
5   RETURN CONCAT('Hello, ',s,'!');
6   END
```

Funktion ausführen:
`SELECT hello('Tomas');`

ArrivalTime Funktion

- Schreiben Sie eine Funktion "ArrivalTime" welche als Input eine Flugnummer (bestehend aus minimal 3 und maximal 8 Zeichen) erwartet und die Ankunftszeit (Abflugzeit + Flugdauer) des Fluges berechnet (zur Vereinfachung ohne Berücksichtigung der verschiedenen Zeitzonen).

```
SELECT f.FlightNo,  
f.DepartureDateAndTimeUTC as DepartureTime,  
ArrivalTime(f.FlightNo) as ArrivalTime  
FROM flightexecution f;
```

	FlightNo	DepartureTime	ArrivalTime
▶	IBE1764	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH1354	2018-12-31 10:30:45	2018-12-31 12:30:45
	LH1769	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH3333	2018-06-05 09:30:45	2018-06-05 12:05:45
	LH3584	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH5301	2018-07-02 16:20:00	2018-07-03 04:40:00
	IBE1684	2018-12-31 10:30:45	2018-12-31 13:50:45
	IBE3843	2018-12-31 10:30:45	2018-12-31 12:30:45
	LH1167	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH1943	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH2301	2018-07-26 10:50:00	2018-07-26 16:30:00
	IBE1846	2018-12-31 10:30:45	2018-12-31 13:10:45
	LH1973	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH7660	2018-12-31 10:30:45	2018-12-31 14:30:45
	IBE4681	2018-12-31 10:30:45	2018-12-31 13:30:45
	LH1761	2018-12-31 10:30:45	2018-12-31 14:30:45
	LH3842	2018-12-31 10:30:45	2018-12-31 12:30:45

ArrivalTime Funktion - Vorschlag

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `ArrivalTime`(fln varchar(8)) RETURNS datetime
2   BEGIN
3     DECLARE arrivalDate DATETIME;
4     DECLARE departureTime DATETIME;
5     DECLARE duration int;
6
7   IF CHAR_LENGTH(flน) >= 3 AND CHAR_LENGTH(flน) <= 8 THEN
8     select f.FlightDurationInMinutes into duration from flightexecution f where FlightNo = flน;
9     select f.DepartureDateAndTimeUTC into departureTime from flightexecution f where FlightNo = flน;
10    select DATE_ADD(departureTime , INTERVAL duration MINUTE) into arrivalDate;
11  ELSE SET arrivalDate = Null;
12  END IF;
13
14  RETURN arrivalDate;
15  END
```

DELIMITER \$\$

```
CREATE FUNCTION `ArrivalTime`(flน varchar(8)) RETURNS datetime
```

```
BEGIN
```

```
    DECLARE arrivalDate DATETIME;
```

```
    DECLARE departureTime DATETIME;
```

```
    DECLARE duration int;
```

```
IF CHAR_LENGTH(flน) >= 3 AND CHAR_LENGTH(flน) <= 8 THEN
```

```
    select f.FlightDurationInMinutes into duration from flightexecution f where FlightNo = flน;
```

```
    select f.DepartureDateAndTimeUTC into departureTime from flightexecution f where FlightNo = flน;
```

```
    select DATE_ADD(departureTime , INTERVAL duration MINUTE) into arrivalDate;
```

```
    ELSE SET arrivalDate = Null;
```

```
END IF;
```

```
    RETURN arrivalDate;
```

```
END
```

Sichten (Views)

Eine View ist eine Relation, die nicht Teil des **konzeptuellen Schemas** der Datenbank ist, sondern dem Benutzer als **virtuelle Relation** zur Verfügung gestellt wird.

Views können nicht gespeichert werden sondern müssen für jede Anfrage, die sie referenziert **neu berechnet** werden.

- **create view** *View-Name* **as**
<Anfrage-Ausdruck>

View (Sicht) Syntax (<https://dev.mysql.com/doc/refman/8.0/en/create-view.html>)

```
1 CREATE
2     [OR REPLACE]
3     [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
4     [DEFINER = { user | CURRENT_USER }]
5     [SQL SECURITY { DEFINER | INVOKER }]
6     VIEW view_name [(column_list)]
7     AS select_statement
8     [WITH [CASCADED | LOCAL] CHECK OPTION]
```

1) Erstellen von View (Sicht):

```
CREATE VIEW flightview AS
SELECT f.FlightNo, f.DepartureDateAndTimeUTC
FROM flightexecution f;
```

2) Verwenden von View (Sicht): `SELECT * FROM flightview;`

	FlightNo	DepartureDateAndTimeUTC
▶	IBE1764	2018-12-31 10:30:45
	LH1354	2018-12-31 10:30:45
	LH1769	2018-12-31 10:30:45
	LH3333	2018-06-05 09:30:45
	LH3584	2018-12-31 10:30:45
	LH5301	2018-07-02 16:20:00
	IBE1684	2018-11-30 12:12:52
	IBE3843	2018-12-31 10:30:45
	LH1943	2018-12-31 10:30:45
	LH2301	2018-07-26 10:50:00
	IBE1846	2018-12-31 10:30:45
	LH1167	2018-11-30 12:38:52
	LH1973	2018-12-31 10:30:45
	LH7660	2018-12-31 10:30:45
	IBE4681	2018-12-31 10:30:45
	LH1761	2018-12-31 10:30:45
	LH3842	2018-12-31 10:30:45

Flughafen Informationsanzeige: *Frankfurt Departure View*

- *View mit dem Namen: frankfurt_departure*
- *Die View soll angeben, welche Flüge starten von Frankfurt am Main mit folgende Attribute: Flight, DepartureTime, Duration, Destination und ArrivalTime.*

	Flight	DepartureTime	Duration	Destination	ArrivalTime
▶	IBE1846	2018-12-31 10:30:45	160	Düsseldorf International	2018-12-31 13:10:45
	IBE4681	2018-12-31 10:30:45	180	Flughafen Rom-Ciampino	2018-12-31 13:30:45
	LH1761	2018-12-31 10:30:45	240	Berlin-Tegel (Otto-Lilienthal)	2018-12-31 14:30:45
	LH1973	2018-12-31 10:30:45	180	Paris-Charles de Gaulle	2018-12-31 13:30:45
	LH2301	2018-07-26 10:50:00	340	Toronto Pearson International Airport	2018-07-26 16:30:00
	LH3333	2018-06-05 09:30:45	155	München (Franz Josef Strauß)	2018-06-05 12:05:45
	LH5301	2018-07-02 16:20:00	740	Toronto Pearson International Airport	2018-07-03 04:40:00

Frankfurt Departure View – Vorschlag 1

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `airport`.`frankfurt_departure` AS
6     SELECT
7         `airport`.`flightexecution`.`FlightNo` AS `Flight`,
8         `airport`.`flightexecution`.`DepartureDateAndTimeUTC` AS `DepartureTime`,
9         `airport`.`flightexecution`.`FlightDurationInMinutes` AS `Duration`,
10        `airport`.`airport`.`AirportName` AS `Destination`,
11        ARRIVALTIME(`airport`.`flightexecution`.`FlightNo`) AS `ArrivalTime`
12    FROM
13        (`airport`.`flightexecution`
14         JOIN `airport`.`airport`)
15    WHERE
16        `airport`.`flightexecution`.`ICAO_Code_Origin` = (SELECT
17            `airport`.`airport`.`ICAO_Code`
18            FROM
19                `airport`.`airport`
20            WHERE
21                `airport`.`airport`.`AirportName` = 'Frankfurt am Main')
22        AND `airport`.`flightexecution`.`ICAO_Code_Destination` = `airport`.`airport`.`ICAO_Code`
```

```
select * from frankfurt_departure;
```

Frankfurt Departure View – Vorschlag 2

```
CREATE VIEW frankfurt_departure AS
SELECT flightexecution.FlightNo AS Flight,
flightexecution.DepartureDateAndTimeUTC AS DepartureTime,
flightexecution.FlightDurationInMinutes AS Duration,
airport.AirportName AS Destination,
flightexecution.DepartureDateAndTimeUTC + INTERVAL flightexecution.FlightDurationInMinutes
MINUTE AS ArrivalTime
FROM flightexecution, airport.airport
WHERE flightexecution.ICAO_Code_Origin = (select airport.ICAO_Code from airport where
airport.AirportName = 'Frankfurt am Main')
and flightexecution.ICAO_Code_Destination = airport.ICAO_Code;
select * from frankfurt_departure;
```

Frankfurt Departure View – Vorschlag 2

Name: frankfurt_departure

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 CREATE
2 ALGORITHM = UNDEFINED
3 DEFINER = `root`@`localhost`
4 SQL SECURITY DEFINER
5 VIEW `airport`.`frankfurt_departure` AS
6 SELECT
7     `airport`.`flightexecution`.`FlightNo` AS `Flight`,
8     `airport`.`flightexecution`.`DepartureDateAndTimeUTC` AS `DepartureTime`,
9     `airport`.`flightexecution`.`FlightDurationInMinutes` AS `Duration`,
10    `airport`.`airport`.`AirportName` AS `Destination`,
11    `airport`.`flightexecution`.`DepartureDateAndTimeUTC` + INTERVAL `airport`.`flightexecution`.`FlightDurationInMinutes` MINUTE AS `ArrivalTime`
12 FROM
13     (`airport`.`flightexecution`
14     JOIN `airport`.`airport`)
15 WHERE
16     `airport`.`flightexecution`.`ICAO_Code_Origin` = (SELECT
17         `airport`.`airport`.`ICAO_Code`
18     FROM
19         `airport`.`airport`
20     WHERE
21         `airport`.`airport`.`AirportName` = 'Frankfurt am Main')
22 AND `airport`.`flightexecution`.`ICAO_Code_Destination` = `airport`.`airport`.`ICAO_Code`
```

Trigger Syntax (<https://dev.mysql.com/doc/refman/8.0/en/create-trigger.html>)

- Trigger ist eine Funktion oder Sammlung von SQL-Anweisungen, die man auf eine Tabelle anwendet.
- Dieser Trigger wird dann automatisch bei einem bestimmten **Event/ Ereignisse** ausgelöst. Dabei kann es um **INSERT-**, **UPDATE-** oder **DELETE-**Anweisungen handeln.

```
1 CREATE
2   [DEFINER = { user | CURRENT_USER }]
3   TRIGGER trigger_name
4   trigger_time trigger_event
5   ON tbl_name FOR EACH ROW
6   [trigger_order]
7   trigger_body
8
9   trigger_time: { BEFORE | AFTER }
10
11  trigger_event: { INSERT | UPDATE | DELETE }
12
13  trigger_order: { FOLLOWS | PRECEDES } other_trigger_name
```

- Trigger-time:
 - **BEFORE** - wird vor der Datenmanipulation ausgelöst und ersetzt die eigentliche Aktion.
 - **AFTER** - wird nach der Datenmanipulation ausgelöst.

Protokoll-Trigger Beispiel

- Protokolliere für jede Update-Aktion auf der Tabelle ***Flightexecution*** ***Action, User, FlightNo, Time*** einer Protokolltabelle ***Flight_Audit***.

(1) Erstelle Protokolltabelle:

```
CREATE TABLE airport.Flight_Audit (  
  Action VARCHAR(50) DEFAULT NULL,  
  User VARCHAR(30) NOT NULL,  
  FlightNo VARCHAR(7) NOT NULL,  
  Time TIMESTAMP NOT NULL);
```

(2) Erstelle ***Before_Flight_Update*** Trigger:

```
CREATE TRIGGER airport.Before_Flight_Update  
  BEFORE UPDATE ON flightexecution  
  FOR EACH ROW  
  
BEGIN  
  
  INSERT INTO airport.Flight_Audit  
  SET Action = 'update',  
  User = USER(),  
  FlightNo = OLD.FlightNo,  
  Time = NOW();  
  
END
```

(3) Beispiel Update-Befehl:

```
UPDATE flightexecution SET flightexecution.FlightDurationInMinutes = 140  
WHERE flightexecution.FlightNo = 'IBE1684';
```

Create Frankfurt Arrival View

- *View mit dem Namen: frankfurt_arrival*
- *Die View soll angeben, welche Flüge noch in Frankfurt am Main landen werden (Sie können rein auf der UTC-Zeit arbeiten und müssen keine Zeitzonen berücksichtigen, bereits gelandete Flüge sollen nicht angezeigt werden). Relevant für Frankfurt am Main ist wieder das Town Attribut für die Adresse des Flughafens.*

	FlightNo	von	nach	Departure	Duration	Landing Time
▶	LH3584	München (Franz Josef Strauß)	Frankfurt am Main	2018-12-31 10:30:45	180	2018-12-31 13:30:45
	LH1769	Flughafen Rom-Ciampino	Frankfurt am Main	2018-12-31 10:30:45	180	2018-12-31 13:30:45

Online-ressourcen

- [Tabellenfunktionen](#) von Prof. Dr. Jens Dittrich
- [Trigger Grundlagen Video](#) von Prof. Dr. Jens Dittrich
- [Folien \(Aufbau von Triggern\) von](#) Prof. Dr. T. Kudraß