ceu-economics-and-business.github.io

DB administration. Queries. – ECBS 5146 SQL and Different Shapes of Data

6–8 minutes

Overview

Teaching: 90 min

Questions

- How do you make sure your data stored, can be accessed only by authorized persons?
- How you query your database to obtain data required for your analytics?

Objectives

- Understanding the options of altering a db
- Introduction to database security
- Understanding datatypes
- Present examples and exercise querying databases

Keywords

#ALTERING DB

#SECURITY

#DATA TYPES

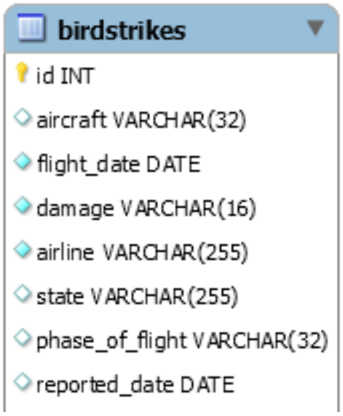
#LOGICAL OPERATORS

#FILTERING

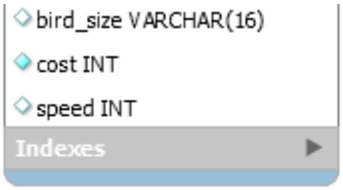
Table of Content

Chapter’s database

No need to load new data, in this chapter we will use only the birdstrikes table loaded in the last chapter:



```
birdstrikes
id INT
aircraft VARCHAR(32)
flight_date DATE
damage VARCHAR(16)
airline VARCHAR(255)
state VARCHAR(255)
phase_of_flight VARCHAR(32)
reported_date DATE
```


bird_size VARCHAR(16)
cost INT
speed INT

Altering your first database

Copy table

```
CREATE TABLE new_birdstrikes LIKE birdstrikes;
SHOW TABLES;
DESCRIBE new_birdstrikes;
SELECT * FROM new_birdstrikes;
```

Delete table

```
DROP TABLE IF EXISTS new_birdstrikes;
```

Create table

> ### Exercise1
>
> Based on the previous chapter, create a table called “employee” with two columns: “id” and “employee_name”. NULL values should not be accepted for these 2 columns.

Insert new rows (records)

Insert lines in employee table one by one

```
INSERT INTO employee (id,employee_name)
VALUES(1,'Student1');
INSERT INTO employee (id,employee_name)
```

```
VALUES(2,'Student2');
INSERT INTO employee (id,employee_name)
VALUES(3,'Student3');
```

Let’s check the results

What happens if you try this (and why)?

```
INSERT INTO employee (id,employee_name)
VALUES(3,'Student4');
```

Updating rows

```
UPDATE employee SET employee_name='Arnold Schwarzenegger' WHERE id = '1';
UPDATE employee SET employee_name='The Other Arnold' WHERE id = '2';
```

Let’s check the results

Deleting rows

Deleting some records

```
DELETE FROM employee WHERE id = 3;
```

Let’s check the results

Deleting all rows

Let’s check the results

Users and privileges

Creating new user

```
CREATE USER 'laszlosallo'@'%' IDENTIFIED BY
'laszlosallo1';
```

Giving full rights for table employee

```
GRANT ALL ON birdstrikes.employee TO
'laszlosallo'@'%';
```

Giving rights to see one column of birdstrikes

```
GRANT SELECT (state) ON birdstrikes.birdstrikes TO
'laszlosallo'@'%';
```

Deleting user

```
DROP USER 'laszlosallo'@'%';
```

More advanced selects

New column

Create a new column

```
SELECT *, speed/2 FROM birdstrikes;
```

Aliasing

```
SELECT *, speed/2 AS halfspeed FROM birdstrikes;
```

Using Limit

List the first 10 records

```
SELECT * FROM birdstrikes LIMIT 10;
```

List the first 1 record, after the the first 10

```
SELECT * FROM birdstrikes LIMIT 10,1;
```

> ### Exercise2
>
> What state figures in the 145th line of our database?

Ordering data

Order by a field

```
SELECT state, cost FROM birdstrikes ORDER BY cost;
```

Order by a multiple fields

```
SELECT state, cost FROM birdstrikes ORDER BY state,
cost ASC;
```

Reverse ordering

```
SELECT state, cost FROM birdstrikes ORDER BY cost
DESC;
```

> ### Exercise3
>
> What is flight_date of the latest birstrike in this database?

Unique values

Of a column

```
SELECT DISTINCT damage FROM birdstrikes;
```

Unique pairs

```
SELECT DISTINCT airline, damage FROM birdstrikes;
```

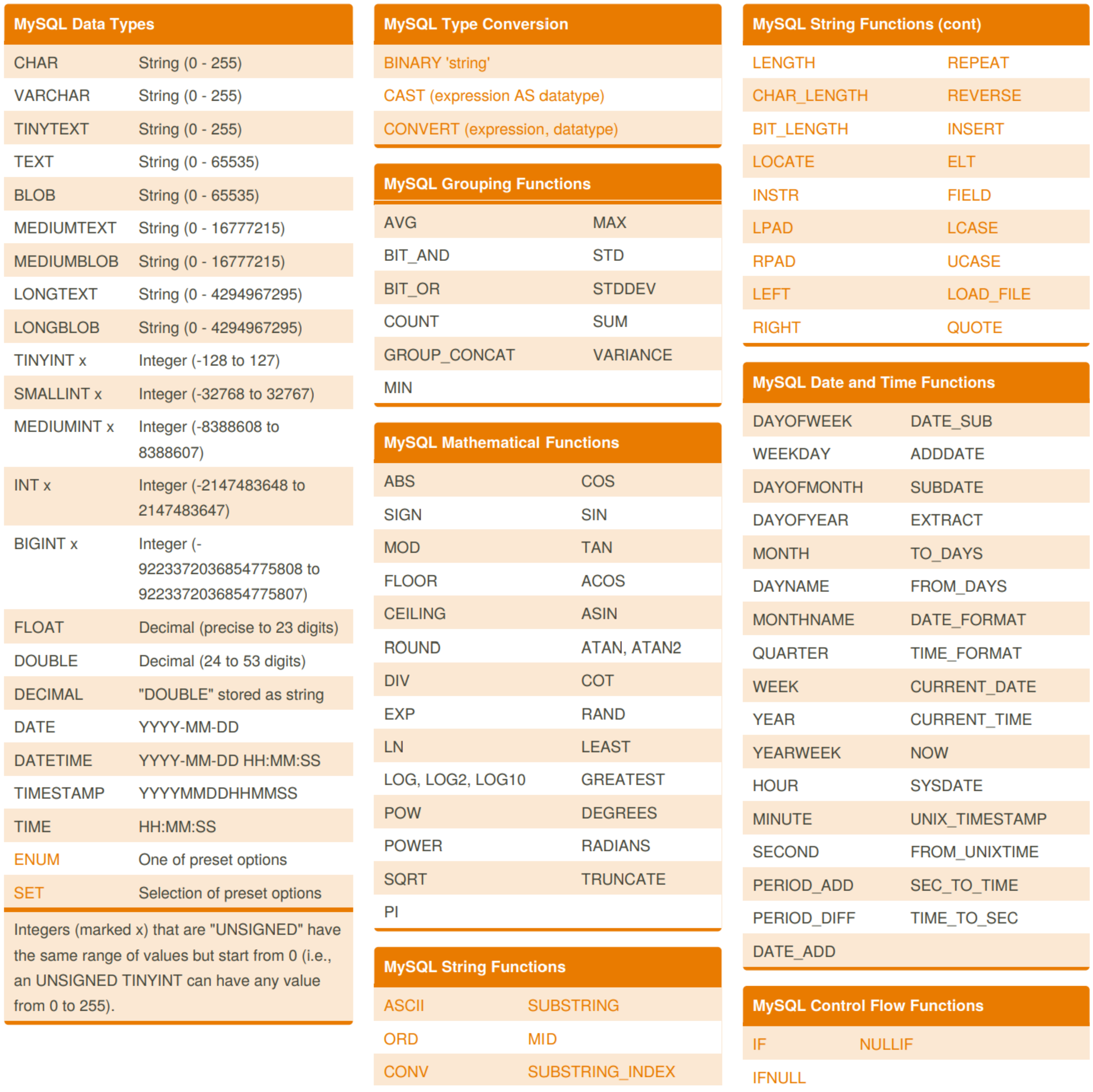
> ### Exercise4
>
> What was the cost of the 50th most expensive damage?

Filtering

Select the lines where states is Alabama

```
SELECT * FROM birdstrikes WHERE state = 'Alabama';
```

Data types



MySQL Data Types	
CHAR	String (0 - 255)
VARCHAR	String (0 - 255)
TINYTEXT	String (0 - 255)
TEXT	String (0 - 65535)
BLOB	String (0 - 65535)
MEDIUMTEXT	String (0 - 16777215)
MEDIUMBLOB	String (0 - 16777215)
LONGTEXT	String (0 - 4294967295)
LONGBLOB	String (0 - 4294967295)
TINYINT x	Integer (-128 to 127)
SMALLINT x	Integer (-32768 to 32767)
MEDIUMINT x	Integer (-8388608 to 8388607)
INT x	Integer (-2147483648 to 2147483647)
BIGINT x	Integer (-9223372036854775808 to 9223372036854775807)
FLOAT	Decimal (precise to 23 digits)
DOUBLE	Decimal (24 to 53 digits)
DECIMAL	"DOUBLE" stored as string
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS
ENUM	One of preset options
SET	Selection of preset options

Integers (marked x) that are "UNSIGNED" have the same range of values but start from 0 (i.e., an UNSIGNED TINYINT can have any value from 0 to 255).

MySQL Type Conversion
BINARY 'string'
CAST (expression AS datatype)
CONVERT (expression, datatype)

MySQL Grouping Functions	
AVG	MAX
BIT_AND	STD
BIT_OR	STDDEV
COUNT	SUM
GROUP_CONCAT	VARIANCE
MIN	

MySQL Mathematical Functions	
ABS	COS
SIGN	SIN
MOD	TAN
FLOOR	ACOS
CEILING	ASIN
ROUND	ATAN, ATAN2
DIV	COT
EXP	RAND
LN	LEAST
LOG, LOG2, LOG10	GREATEST
POW	DEGREES
POWER	RADIANS
SQRT	TRUNCATE
PI	

MySQL String Functions	
ASCII	SUBSTRING
ORD	MID
CONV	SUBSTRING_INDEX

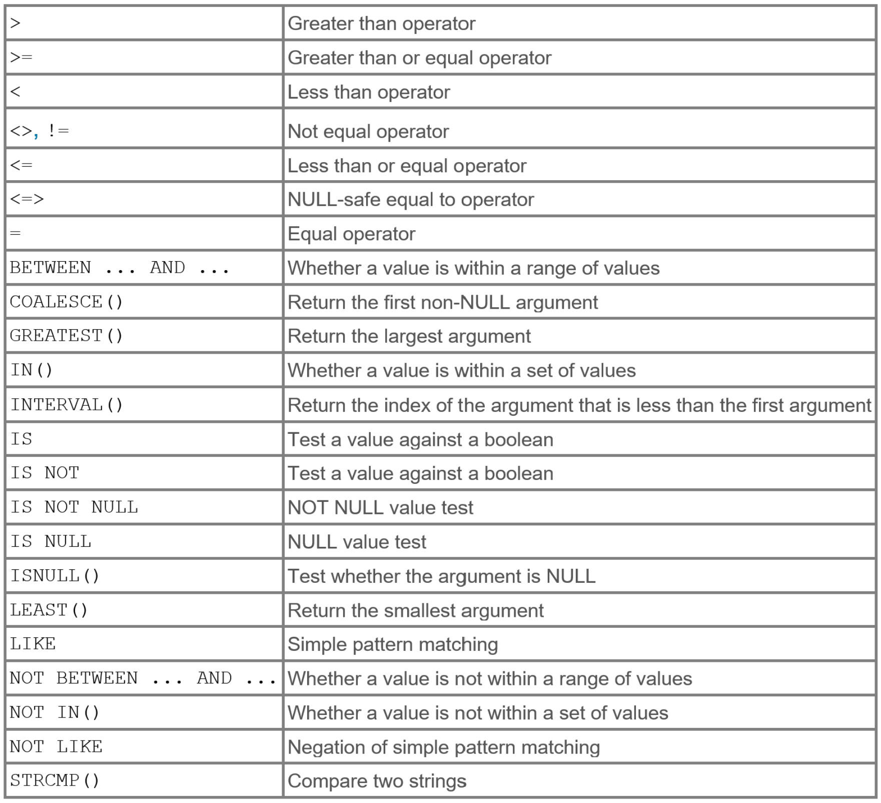
MySQL String Functions (cont)	
LENGTH	REPEAT
CHAR_LENGTH	REVERSE
BIT_LENGTH	INSERT
LOCATE	ELT
INSTR	FIELD
LPAD	LCASE
RPAD	UCASE
LEFT	LOAD_FILE
RIGHT	QUOTE

MySQL Date and Time Functions	
DAYOFWEEK	DATE_SUB
WEEKDAY	ADDDATE
DAYOFMONTH	SUBDATE
DAYOFYEAR	EXTRACT
MONTH	TO_DAYS
DAYNAME	FROM_DAYS
MONTHNAME	DATE_FORMAT
QUARTER	TIME_FORMAT
WEEK	CURRENT_DATE
YEAR	CURRENT_TIME
YEARWEEK	NOW
HOUR	SYSDATE
MINUTE	UNIX_TIMESTAMP
SECOND	FROM_UNIXTIME
PERIOD_ADD	SEC_TO_TIME
PERIOD_DIFF	TIME_TO_SEC
DATE_ADD	

MySQL Control Flow Functions	
IF	NULLIF
IFNULL	

BIN	LTRIM
OCT	RTRIM
HEX	TRIM
CHAR	SOUNDEX
CONCAT	SPACE
CONCAT_WS	REPLACE

Comparison Operators



>	Greater than operator
>=	Greater than or equal operator
<	Less than operator
<>, !=	Not equal operator
<=	Less than or equal operator
<=>	NULL-safe equal to operator
=	Equal operator
BETWEEN ... AND ...	Whether a value is within a range of values
COALESCE()	Return the first non-NULL argument
GREATEST()	Return the largest argument
IN()	Whether a value is within a set of values
INTERVAL()	Return the index of the argument that is less than the first argument
IS	Test a value against a boolean
IS NOT	Test a value against a boolean
IS NOT NULL	NOT NULL value test
IS NULL	NULL value test
ISNULL()	Test whether the argument is NULL
LEAST()	Return the smallest argument
LIKE	Simple pattern matching
NOT BETWEEN ... AND ...	Whether a value is not within a range of values
NOT IN()	Whether a value is not within a set of values
NOT LIKE	Negation of simple pattern matching
STRCMP()	Compare two strings

Filtering with VARCHAR

NOT EQUAL

Select the lines where states is not Alabama

```
SELECT * FROM birdstrikes WHERE state != 'Alabama'
```

States starting with ‘A’

LIKE

```
SELECT DISTINCT state FROM birdstrikes WHERE state LIKE 'A%';
```

Note the case (in)sensitivity

```
SELECT DISTINCT state FROM birdstrikes WHERE state LIKE 'a%';
```

States starting with ‘ala’

```
SELECT DISTINCT state FROM birdstrikes WHERE state LIKE 'ala%';
```

States starting with ‘North ‘ followed by any character, followed by an ‘a’, followed by anything

```
SELECT DISTINCT state FROM birdstrikes WHERE state LIKE 'North _a%';
```

States not starting with ‘A’

```
SELECT DISTINCT state FROM birdstrikes WHERE state NOT LIKE 'a%' ORDER BY state;
```

Logical operators

Filter by multiple conditions

```
SELECT * FROM birdstrikes WHERE state = 'Alabama' AND bird_size = 'Small';
SELECT * FROM birdstrikes WHERE state = 'Alabama' OR state = 'Missouri';
```

IS NOT NULL

Filtering out nulls and empty strings

```
SELECT DISTINCT state FROM birdstrikes WHERE state IS
NOT NULL AND state != '' ORDER BY state;
```

IN

What if I need ‘Alabama’, ‘Missouri’,’New York’,’Alaska’? Should we concatenate 4 AND filters?

```
SELECT * FROM birdstrikes WHERE state IN ('Alabama',
'Missouri','New York','Alaska');
```

LENGTH

Listing states with 5 characters

```
SELECT DISTINCT(state) FROM birdstrikes WHERE
LENGTH(state) = 5;
```

Filtering with INT

Speed equals 350

```
SELECT * FROM birdstrikes WHERE speed = 350;
```

Speed equal or more than 25000

```
SELECT * FROM birdstrikes WHERE speed >= 10000;
```

ROUND, SQRT

```
SELECT ROUND(SQRT(speed/2) * 10) AS synthetic_speed
FROM birdstrikes;
```

BETWEEN

```
SELECT * FROM birdstrikes where cost BETWEEN 20 AND
40;
```

> ## Exercise5
>
> What state figures in the 2nd record, if you filter out all records which have no state and no bird_size specified?

Filtering with DATE

Date is “2000-01-02”

```
SELECT * FROM birdstrikes WHERE flight_date =
"2000-01-02";
```

All entries where flight_date is between “2000-01-01” AND “2000-01-03”

```
SELECT * FROM birdstrikes WHERE flight_date >=
'2000-01-01' AND flight_date <= '2000-01-03';
```

BETWEEN

```
SELECT * FROM birdstrikes where flight_date BETWEEN
"2000-01-01" AND "2000-01-03";
```

> ## Exercise6
>
> How many days elapsed between the current date and the flights happening in week 52, for incidents from Colorado? (Hint: use NOW, DATEDIFF, WEEKOFYEAR)

> ## Homework 2

> - Upload the solution of exercise 1-6 to your GitHub repo in a folder called HW2
> - Make sure to submit both the SQL statements and answers to the questions
> - The required data format for submission is a .sql file
> - Submit GitHub repo link to moodle when you are ready