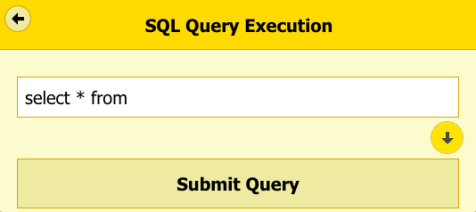
# The data of BeeBIT

Every minute, sensors placed within the eHive and sensors monitoring its surroundings collect data. All data are digitalised and stored on a central computer, a so-called server. Using the language of the database (SQL), we can fetch and display the data we are interested in.

Go to http://www.beebit.de/diagram/ to formulate a question, or query, you want answered by the database.

## Task 1

1. Familiarise yourself with the database: Use SHOW TABLES to display all the tables available in the database.
2. Every eHive collects data on the bees and on the weather. Take a look at the table labelled *sensor* and find out how many sensors belong to the beehive, the weather station, and the electronic equipment, respectively.

SELECT \* FROM sensor

eHive: 10 weather station: 8 electronic equipment: 5

1. Select those sensors whose *name* contains the word “temperature”. To do this, add the following line:

WHERE name LIKE “%Temperature%”

Hint: The LIKE operator is used in a WHERE clause to search for similar entries – the percentage sign (%) is used as a wildcard for any number of characters.

SELECT \*

FROM sensor

WHERE name LIKE “%Temperature%”

number of temperature sensors: 8

1. What is the *sensor\_id* of the sensor monitoring the outside temperature?

sensor\_id: 128

## Task 2

To get information of a particular eHive, you need to understand the structure of the database:

1. Take a look at the table labelled *beebox*:
   1. How many eHives are there in total and how many of them are situated in Vienna?

SELECT \* FROM beebox

insgesamt: 12 Wien: 3

* 1. What is the *beebox\_id* of Deutschhaus Gymnasium Wuerzburg?

beebox\_id: 1

1. All recorded data are stored in table *datalayer0*. Write down the identifiers (names) of all the columns in *datalayer0* and think about what they may signify.

Hint: This query will probably take very long and force a timeout as it queries ALL ENTRIES of the database. Add the LIMIT command (for example: LIMIT 10) to the SQL query to display only a limited number of rows.

timestamp: date and time of the measurement – in “UNIX time”

beeboxsensor\_id: ID of one sensor of a specific eHive

value: a measured value

1. Each sensor in the database is assigned a unique number, the *beeboxsensor\_id*, which identifies it unambiguously. The table labelled *beeboxsensor* shows which eHive (*beebox\_id*) the sensor belongs to and which type of sensor it is (*sensor\_id*).   
   Use the previous results to determine the *beeboxsensor\_id* for the sensor measuring the outside temperature in Wuerzburg and write down the query that you used.

beeboxsensor\_id: 13

SELECT beeboxsensor\_id

FROM beebox

WHERE beebox\_id = 1

AND sensor\_id = 128

1. Now take a closer look at the temperatures of Wuerzburg: Find the maximum temperature and the minimum temperature measured in Wuerzburg.

Hint:Sort the query by temperature (ORDER BY % [DESC]) and limit it (LIMIT).

maximum temperature: \_\_\_\_\_\_°C

minimum temperature: \_\_\_\_\_\_°C

SELECT MAX(value) // SELECT MIN(value)

FROM datalayer0

WHERE beeboxsensor\_id = 13

1. Repeat the steps above for at least one other eHive and compare your results.