

## Prosjekt-leveranse 2 / Project delivery 2

Changes in our database topology from part 1:

- We removed the Kaffestatistikk table/entity, we found that it doesn't really make sense to have it as a separate table. We added this attribute under the 'Brukere/User' entity.
- We removed the double primary keys in RoastedCoffee and Batch, and we now only have one primary key in both of those tables, they are: CoffeeID and BatchID

b) describe the application and how it solves each user story

The application is in the files 'coffeeDBtables.py', and 'coffee.py'. Running the 'coffeeDBtables' file makes a 'coffee.db' database file with sqlite3. The different tables in coffee.db are being made from cursor.execute commands in the 'coffeeDBtables' file. When this script is executed the database file is made, we can connect to it with our other script, 'coffee.py'.

This coffee.py file asks the user which user story it wants to fulfill, with a number between 1 and 5. While the user chooses a number between 1-5, the file will continue asking if you want to read a user story or quit. If you press 'q' instead of a number 1-5, the while loop will break and close the database connection.

### User story 1:

For the user input data, we queried the user in python for input like this: `attribute(e.g coffee_name)=input("name of attribute")`. We then added these values to their respective entities/tables with `cursor.execute('INSERT INTO table VALUES(attribute)')`.

The attributes that were not asked for as user input, but still were in the story, was added manually to the database by typing multiple `cursor.execute('INSERT INTO table VALUES('values')')`. The attributes that aren't mentioned in the story at all were set to 'NULL'.

### User story 2:

```
SELECT FullName, COUNT(DISTINCT CoffeeID) AS TasteCount
FROM User NATURAL JOIN CoffeeTasting
WHERE TasteDate LIKE '2022%'
GROUP BY FullName
ORDER BY TasteCount DESC
```

In user story 2, a user should be able to print a list of the users who have tasted most unique coffees so far this year. We solved this by writing a sql query in python. The query selects the

24.03.22

name, and how many distinct coffees have been tasted. To match a user to a coffee tasting, we joined the 'user' table, with 'CoffeeTasting' on where the 'email' attribute matches in both tables. To match the person and their coffeetasting with a coffee type, we joined again with 'RoastedCoffee' table where the 'Roastery' and 'CoffeeName' attributes match. To get only the list of tasted coffees from this year, we sorted by using: 'WHERE TasteDate LIKE '%2022''. This means that the date of tasting can be any date and month, but the year is equal to 2022. We group by email in order to separate people, then we sort by the number of distinct coffees tasted in descending order.

**Textual output:**

('Jakob Tøssebro', 1)  
 ('Kristian Lynes', 1)

**User story 3:**

```
SELECT Roastery.RoasteryName, RoastedCoffee.CoffeeName, RoastedCoffee.KiloPrice,
AVG(CoffeeTasting.Points) AS AVGPoints
FROM (Roastery JOIN RoastedCoffee
ON (Roastery.RoasteryName = RoastedCoffee.RoasteryName)
JOIN CoffeeTasting ON (CoffeeTasting.CoffeeID = RoastedCoffee.CoffeeID))
ORDER BY AVGPoints
```

**Textual output:**

('The Ugandan Roast', 'Ugandan Light', 420, 7.8)

**User story 4:**

```
SELECT Roastery.RoasteryName, RoastedCoffee.CoffeeName
FROM (Roastery JOIN RoastedCoffee
ON (RoastedCoffee.RoasteryName = Roastery.RoasteryName))
JOIN CoffeeTasting ON (RoastedCoffee.CoffeeID = CoffeeTasting.CoffeeID)
WHERE Notes LIKE '%floral%' OR CoffeeDescription = '%floral%'
```

**Textual output:**

('Trondheimsbrenneriet', 'Tronderbonna 2021')

24.03.22

**User story 5:**

```
SELECT Roastery.RoasteryName, RoastedCoffee.CoffeeName
FROM Farm JOIN Batch ON (Farm.FarmName = Batch.Farm)
JOIN RoastedCoffee ON (RoastedCoffee.BatchID = Batch.BatchID)
JOIN Roastery ON (Roastery.RoasteryName = RoastedCoffee.RoasteryName)
WHERE Batch.ProcessingMethod <> 'washed' AND Farm.Country = 'Rwanda'
OR Batch.ProcessingMethod <> 'washed' AND Farm.Country = 'Colombia'
```

**Textual output:**

```
('The Ugandan Roast', 'Ugandan Light')
('Jacobsen & Svart', 'Vinterkaffe 2022')
```