Datastructure:

```
User:           {
                        uid: int,
                        fname: str,
                        lname: str,
                        login: str,
                        pw: str
                }


Questions:      {
                        qid: int,
                        topic: str,
                        question: str,
                        answers: list[str],
                        correct_index: int,
                        difficulty: int
                }


Progress:       {
                        uid: int,
                        Machine Learning: [{ timestamp: int
                                             num_questions: int
                                             correct: int }],
                        ...
                        badges: list[{bid: int,  count: int}]
                }


Badges:         {
                        bid: int,
                        name: str,
                        target: int
                }
```

Login credentials for example users:

login	pw
mm1	mm1
mr1	mr1
jm1	jm1

```
db.get_user(login: str, pw: str) -> dict
        requires login and pw from user.
        returns dict with ‘fname’, ‘lname’ and ‘uid’.

db.get_questions(topic: str) -> list[dict]
        requires topic.
        returns list of all question dicts from given topic.

db.get_questions(topic: str, num: int) -> list[dict]
        requires topic and number.
        returns list of num random question dicts from given topic.

db.get_questions(qid: int) -> dict
        requires question id.
        returns specific question dict for that id.

db.get_progress(uid: int) -> dict
        requires user id.
        returns progress dict from given user.

db.get_badges() -> list[dict]
        returns list of all badge dicts.

db.get_badges(bid: int) -> dict
        requires badge id.
        returns specific badge dict for given id.

db.set_questions(q_dict: dict)
        requires some parts of question dict:
                for update:
                        mandatory:      ‘qid’, ‘answered’
                        optional:       ‘topic’, ‘question’, ‘answers’, ‘correct_index’, ‘difficulty’
                for new questions:
                        mandatory:      ‘topic’, ‘question’, ‘answers’, ‘correct_index’, ‘difficulty’
                        optional:       ‘answered’

db.set_progress(p_dict: dict)
        requires some parts of progress dict:
                for update and creating new progress:
                        mandatory:      ‘uid’
                        optional:       topics, ‘badges’

db.set_badge(b_dict: dict)
        requires some parts of badge dict:
                for update:
                        mandatory:      ‘bid’
                        optional:       ‘name’, ‘target’
                for new badge:
                        mandatory:      ‘name’, ‘target’
```

db.update_progress(uid: int, quiz_results: dict)

requires user id and quiz results as dict:

```
quiz_results:   { topic: str,
                  num_questions: int,
                  correct: int }
```

db.get_user_statistics(uid: int)

requires user id.

returns y-axes per topic with percentage of correct answers for each completed quiz.