Silvio Venturino 01601
Catello Staiano 01602

Q&A platform

Basi dati II

Professore Biasi
Professoressa Tortora

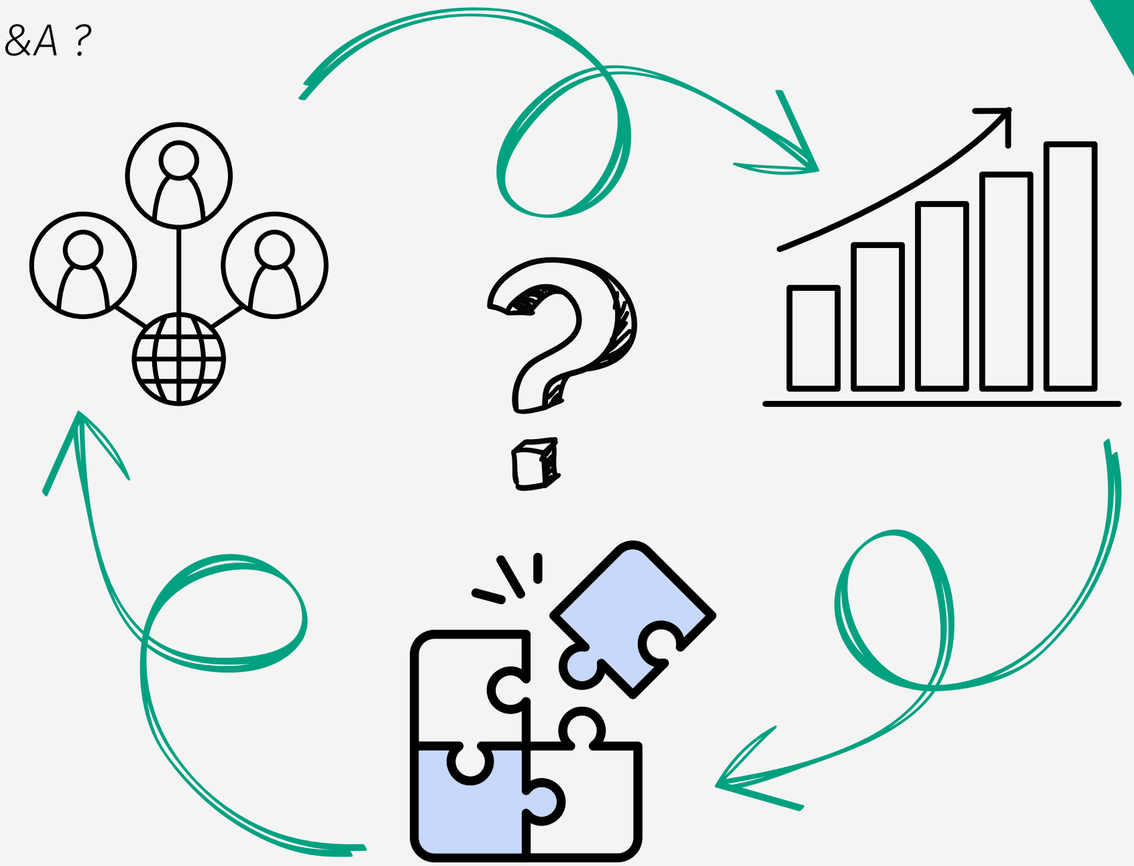
Programma

- **Introduzione**: Cos’è Q&A
  - **Requisiti funzionali**
- **Fasi realizzazione DB**
  - **Raccolta dati**
  - **Data cleaning**
  - **Import database**
- **Tecnologie utilizzate**
  - **MongoDB**
  - **Python, Flask**
- **Query**

Silvio Venturino 01601
Catello Staiano 01602

Introduzione

- *Cos'è Q&A ?*



Requisiti funzionali

Registrazione & Login

Inserimento Domanda & Risposta

Visualizzazione & Ricerca Domande

Aggiornamento & Eliminazione Domande

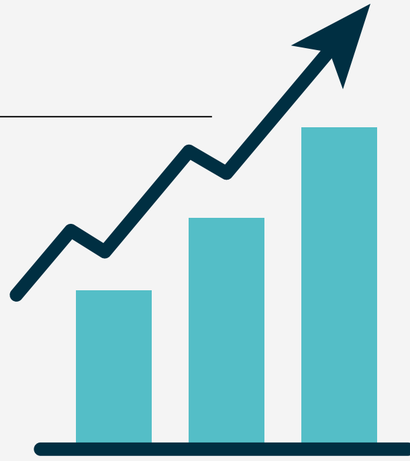
Rating Domande

Fasi realizzazione DB

Raccolta dati

Data cleaning

Import database



Raccolta dati

Download Dataset

link: https://www.kaggle.com/datasets/stackoverflow/

Data cleaning

- Estrapolazione di 2 CSV dal dataset: **answers.cvs** e **questions.csv**
- Aggiunta collezione **users.csv**
- Rimozione campi inutili: is AcceptedAnswer
- Modifica del nome del campo: da “ParentId” a “QuestionId” nel file Answers.csv
- Conversione di data e ora nel formato GG/MM/AAAA - HH/mm/ss

Import database e collezioni

mongoDB

Users

```
{
_id: <ObjectId>
Id: <int>
Email: <string>
Username: <string>
Password: <string>
Created_at <string>
}
```

Questions

```
{
_id: <ObjectId>
Id: <int>
OwnerUserId: <int>
CreationDate: <string>
Score: <int>
Title: <string>
Body: <string>
}
```

Answers

```
{
_id: <ObjectId>
Id: <int>
OwnerUserId: <int>
QuestionId: <int>
CreationDate: <string>
Score: <int>
Body: <string>
}
```

Silvio Venturino 01601
Catello Staiano 01602

Tecnologie utilizzate

Scelta del DBMS

Prestazioni & Scalabilità

- Alta disponibilità (es Indici)
- Sharding del database

Schema flessibile

- Nessuna migrazione in caso di modifiche al database

Supporto per query complesse

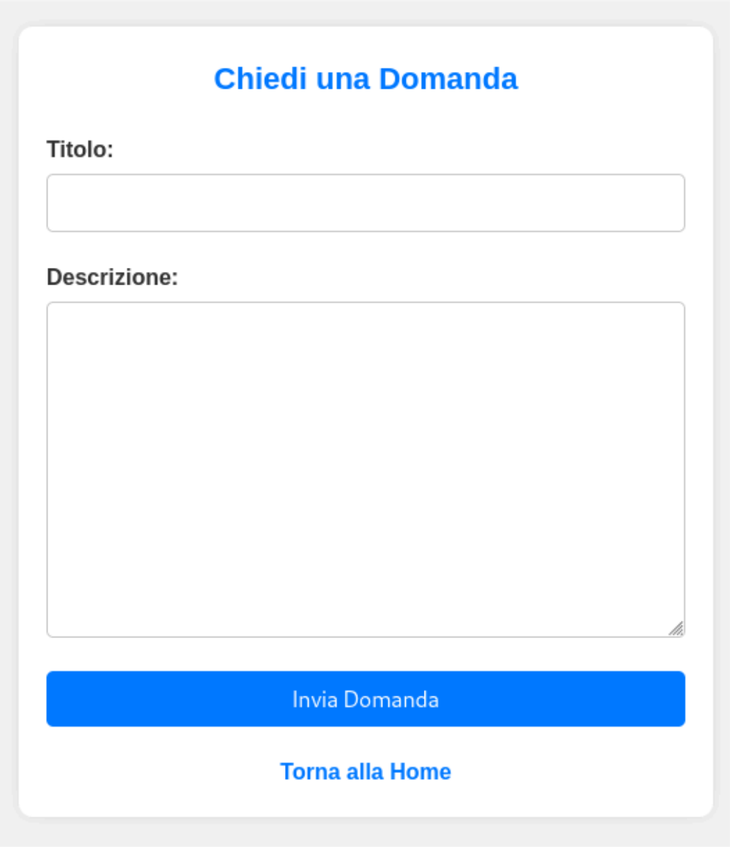
- Aggregazioni e indici permettono di recuperare i dati in pochi passi e con una sintassi facilmente comprensibile

Silvio Venturino 01601
Catello Staiano 01602

Query

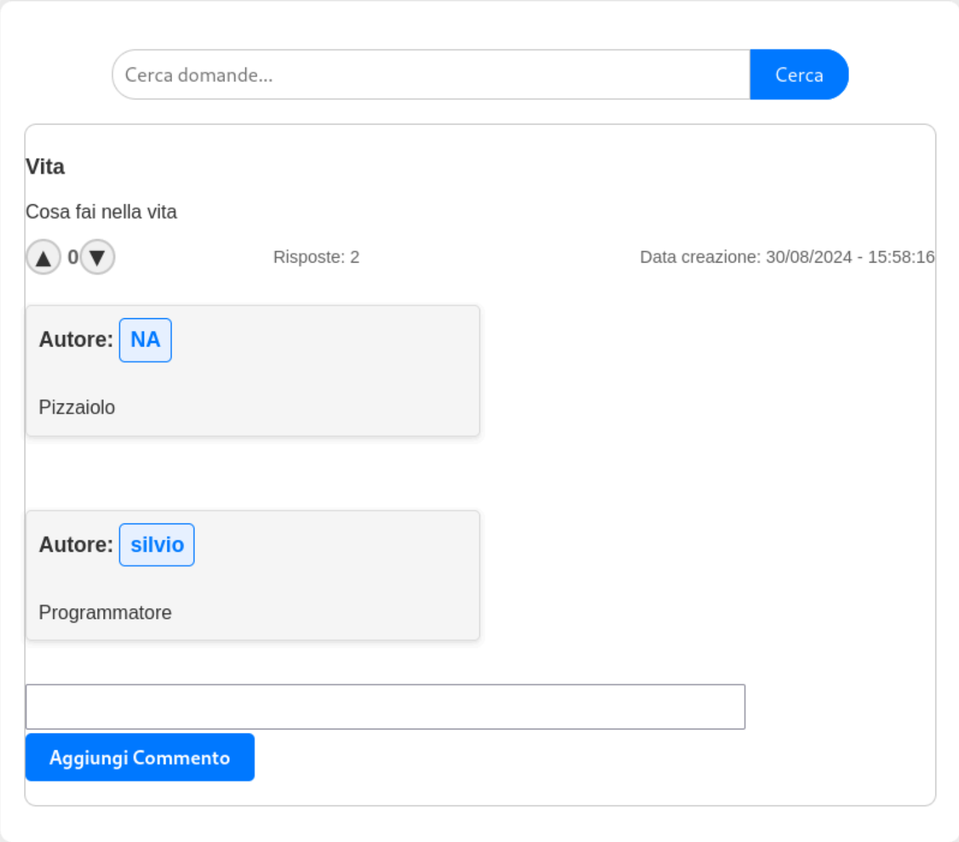
- Inserimento utente e prelievo
- **Inserimento domanda**
- **Visualizzazione Domanda con commenti**
- **Ricerca di una domanda**
- Inserimento di un commento
- **Aggiornamento del rating di una domanda**
- Aggiornamento di una domanda
- **Cancellazione di una domanda di un utente**

Inserimento domanda




```python
@app.route( rule: "/ask", methods=["GET", "POST"])
def ask_question():
    '''Allow users to ask a question'''
    if request.method == "POST":
        title = request.form.get("title")
        body = request.form.get("body")
        if not title or not body:
            return jsonify({'success': False, 'error': "Il titolo e la descrizione sono obbligatori."})
        # Recupera l'ID dell'utente loggato, se presente
        username = session.get("username")
        if username:
            user = users_collection.find_one({"username": username})
            if user:
                owner_user_id = user["Id"]
            else:
                owner_user_id = "NA"
        else:
            owner_user_id = "NA"
        question_id = generate_unique_question_id()
        question = {
            "Id": question_id,
            "Title": title,
            "Body": body,
            "OwnerUserId": owner_user_id,
            "CreationDate": datetime.now().strftime("%d/%m/%Y - %H:%M:%S"),
            "Score": 0
        }
        questions_collection.insert_one(question)
```

Visualizza domande con commenti




```python
@app.route( rule: "/question/<question_id>", methods=['GET','POST'])
def show_question(question_id):
    '''show single question with comments'''
    user_session = users_collection.find_one({"username": session.get("username")}) if "username" in sess
    pipeline = [
        {
            "$match": {
                "Id": int(question_id)
            }
        },
        {
            "$lookup": {
                "from": "answers",  # Collezione di destinazione
                "localField": "Id",  # Campo della collezione `questions`
                "foreignField": "QuestionId",  # Campo della collezione `answers`
                "as": "answers"  # Si aggiungono le answers come oggetto embedded nel documento question
            }
        }
    ]
    questions_with_comments = list(questions_collection.aggregate(pipeline))
```

Array di risposte per una domanda

Ricerca domande (con indice)

linux

Cerca

How to execute linux commands from R via bash under the Windows Subsystem for Linux (WSL)?

The WSL on Windows 10 allows execution of Linux commands and command-line tools via bash.exe. Very usefully, a Linux tool/command can be called from the Windows command-line (`cmd.exe`) by passing it as an argument to bash.exe like so:

```
bash.exe -c <linux command>
```

This is very useful because it should allow Windows-based scripts to combine Windows and Linux tools seamlessly.

Unfortunately, I have failed to call Linux commands from an R script (see below).

0) System

Win10 x64 + Anniversary Update + WSL installed

1) Comparison cases where calling Linux commands work

The following all work for me; shown here just with an example call to `ls`.

- from the windows command-line (cmd.exe prompt)

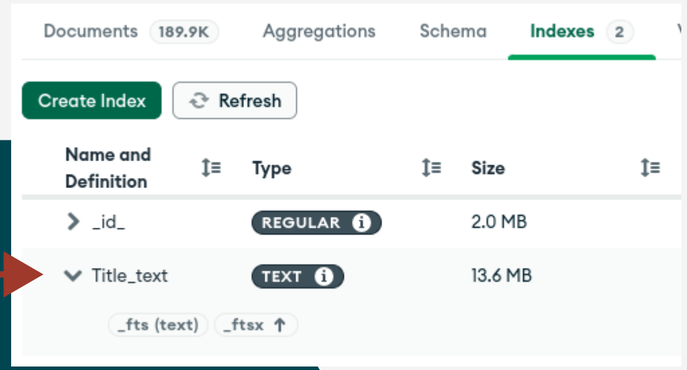
```
bash -c "ls /mnt/a"
bash -c "ls /mnt/a > /mnt/a/test.txt"
```

```python
def search_questions(query):
    # query_no_index = questions_collection.find({'Title': {'$regex': query, '$options': 'i'}}).explain()
    # query_index = questions_collection.find({"$text": {"$search": query}}).explain()
    query_search= questions_collection.find({"$text": {"$search": query}})
    return list(query_search)


Catello +1
@app.route( rule: '/search', methods=['POST'])
def search():
    '''Allow user to search a question by title'''
    query = request.form.get('query').strip()  # Rimuove gli spazi bianchi
    if not query:  # Verifica se la query è vuota
        return render_template( template_name_or_list: 'questions/search_results.html', questions=[], query=query,
                               error="Inserisci una query per cercare.")

    questions = search_questions(query)
    # print(f"domande ricercate: {questions}")
    # Recupera l'utente loggato, se presente
    user_session = users_collection.find_one({"username": session.get("username")}) if "username" in session

    return render_template( template_name_or_list: 'questions/search_results.html', questions=questions, query=que
```



Ricerca domande: prestazioni

Con indice

{'explainVersion': '1', 'queryPlanner': {'namespace': 'qa_platform.questions', 'indexFilterSet': False, 'parsedQuery': {'$text': {'$search': 'linux', '$language': 'english', '$caseSensitive': False, '$diacriticSensitive': False}}, 'queryHash': 'A06CA4BC', 'planCacheKey': '2CF8383A', 'maxIndexedOrSolutionsReached': False, 'maxIndexedAndSolutionsReached': False, 'maxScansToExplodeReached': False, 'winningPlan': {'stage': 'TEXT_MATCH', 'indexPrefix': {}, 'indexName': 'Title_text', 'parsedTextQuery': {'terms': ['linux'], 'negatedTerms': [], 'phrases': [], 'negatedPhrases': []}, 'textIndexVersion': 3, 'inputStage': {'stage': 'FETCH', 'inputStage': {'stage': 'IXSCAN', 'keyPattern': {'_fts': 'text', '_ftsx': 1}, 'indexName': 'Title_text', 'isMultiKey': True, 'isUnique': False, 'isSparse': False, 'isPartial': False, 'indexVersion': 2, 'direction': 'backward', 'indexBounds': {}}}}, 'rejectedPlans': []}, 'executionStats': {'executionSuccess': True, 'nReturned': 372, 'executionTimeMillis': 241, 'totalKeysExamined': 372, 'totalDocsExamined': 372, 'executionStages': {'stage': 'TEXT_MATCH', 'nReturned': 372, 'executionTimeMillisEstimate': 240, 'works': 373, 'advanced': 372, 'needTime': 0, 'needYield': 0, 'saveState': 12, 'restoreState': 12, 'isEOF': 1, 'indexPrefix': {}, 'indexName': 'Title_text', 'parsedTextQuery': {'terms': ['linux'], 'negatedTerms': [], 'phrases': [], 'negatedPhrases': []}, 'textIndexVersion': 3, 'docsRejected': 0, 'inputStage': {'stage': 'FETCH', 'nReturned': 372, 'executionTimeMillisEstimate': 240, 'works': 373, 'advanced': 372, 'needTime': 0, 'needYield': 0, 'saveState': 12, 'restoreState': 12, 'isEOF': 1, 'docsExamined': 372, 'alreadyHasObj': 0, 'inputStage': {'stage': 'IXSCAN', 'nReturned': 372, 'executionTimeMillisEstimate': 0, 'works': 373, 'advanced': 372, 'needTime': 0, 'needYield': 0, 'saveState': 12, 'restoreState': 12,

Senza indice

{'explainVersion': '1', 'queryPlanner': {'namespace': 'qa_platform.questions', 'indexFilterSet': False, 'parsedQuery': {'Title': {'$regex': 'linux', '$options': 'i'}}, 'queryHash': '00412102', 'planCacheKey': '00412102', 'maxIndexedOrSolutionsReached': False, 'maxIndexedAndSolutionsReached': False, 'maxScansToExplodeReached': False, 'winningPlan': {'stage': 'COLLSCAN', 'filter': {'Title': {'$regex': 'linux', '$options': 'i'}}, 'direction': 'forward'}, 'rejectedPlans': []}, 'executionStats': {'executionSuccess': True, 'nReturned': 376, 'executionTimeMillis': 1726, 'totalKeysExamined': 0, 'totalDocsExamined': 189932, 'executionStages': {'stage': 'COLLSCAN', 'filter': {'Title': {'$regex': 'linux', '$options': 'i'}}, 'nReturned': 376, 'executionTimeMillisEstimate': 891, 'works': 189933, 'advanced': 376, 'needTime': 189556, 'needYield': 0, 'saveState': 192, 'restoreState': 192, 'isEOF': 1, 'direction': 'forward', 'docsExamined': 189932}, 'allPlansExecution': []}, 'command': {'find': 'questions', 'filter': {'Title': {'$regex': 'linux', '$options': 'i'}}, '$db': 'qa_platform'}, 'serverInfo': {'host': 'fedora', 'port': 27017, 'version': '7.0.12', 'gitVersion': 'b6513ce0781db6818e24619e8a461eae90bc94fc'}, 'serverParameters': {'internalQueryFacetBufferSizeBytes': 104857600, 'internalQueryFacetMaxOutputDocSizeBytes': 104857600, 'internalLookupStageIntermediateDocumentMaxSizeBytes': 104857600, 'internalDocumentSourceGroupMaxMemoryBytes': 104857600, 'internalQueryMaxBlockingSortMemoryUsageBytes': 104857600, 'internalQueryProhibitBlockingMergeOnMongoS': 0, 'internalQueryMaxAddToSetBytes': 104857600, 'internalDocumentSourceSetWindowFieldsMaxMemoryBytes': 104857600,

Aggiornamento del rating di una domanda

How to upgrade R in linux?

I am new in linux. I am using linux mint 18.1 . I have installed R using system software manager. My current R version is 3.2 . But I want to upgrade it to version 3.4. Need help

▲ 0 ▼

How to upgrade R in linux?

How to upgrade R in linux?

I am new in linux. I am using linux mint 18.1 . I have installed R using system software manager. My current R version is 3.2 . But I want to upgrade it to version 3.4. Need help

▲ 1 ▼

How to upgrade R in linux?

```python
@app.route('/update_score', methods=['POST'])
def update_score():
    data = request.get_json()
    question_id = data.get('questionId')
    vote_type = data.get('voteType')
    # Validazione dei dati
    if not question_id or not isinstance(vote_type, int):
        return jsonify({'success': False, 'error': 'Dati invalidi'}), 400
    try:
        question_id = int(question_id)
    except ValueError:
        return jsonify({'success': False, 'error': 'ID domanda non valido'}), 400
    result = questions_collection.update_one(
        {'Id': question_id},
        {'$inc': {'Score': vote_type}}
    )
    if result.matched_count == 0:
        return jsonify({'success': False, 'error': 'Domanda non trovata'}), 404
    # Recupera il nuovo punteggio aggiornato
    updated_question = questions_collection.find_one({'Id': question_id}, {'Score': 1})
    new_score = updated_question.get('Score', 0)
    return jsonify({'success': True, 'new_score': new_score}), 200
```

Eliminazione di una domanda

```python
silvio2804 *
@app.route( rule: '/delete_question/<int:question_id>', methods=['POST'])
def delete_question(question_id):
    try:
        result = questions_collection.delete_one({'Id': question_id})
        answers_collection.delete_many({'QuestionId': question_id})
        if result.deleted_count == 1:
            flash( message: 'Domanda eliminata con successo!', category: 'success')
        else:
            flash( message: 'Domanda non trovata.', category: 'danger')
    except Exception as e:
        flash( message: f'Errore durante l\'eliminazione della domanda: {str(e)}',

    return redirect(url_for('show_user_questions'))
```

Grazie per l’attenzione