Python Package Metadata Management

Nguyễn Gia Phong—BI9-184

Nguyễn Quốc Thông—BI9-214

Nguyễn Văn Tùng—BI9-229

Trần Minh Vương—BI9-239

University of Science and Technology of Hà Nội

July 8, 2020

Contents

Python
Package
Metadata
Management

Group 8

Introduction
User
Requirements
Data
Definition
Data Query
Conclusion

1 Introduction

2 User Requirements

3 Data Definition
   Entity Relationship Diagram
   Database Schema

4 Data Query

5 Conclusion

Python
Package
Metadata
Management

Group 8

Why?

- Python package managers download whole packages just for metadata
- Mirroring PyPI is expensive (6 TB)
- Middle approach: Mirroring metadata

1 Introduction

2 User Requirements

3 Data Definition
    Entity Relationship Diagram
    Database Schema

4 Data Query

5 Conclusion

- `list_projects()`
  List of registered project names.

- `project_releases(project)`
  List of releases for given `project`, ordered by version.

- `project_release_latest()`
  Latest release of given `project`.

- `belong_to(name)`
  List of projects whose author is `name`.

- `browse(classifier)`: List of (`project`, `version`) of all releases classified by `classifier`.

- `release_data(project, version)`: Metadata of given release: project, version, homepage, author, author's email, summary, keywords, classifiers and dependencies

- `search_name(pattern)`: List of (`project`, `version`, `summary`) where `name` matches `pattern`.

- `search_summary(pattern)`: List of (`project`, `version`, `summary`) where `summary` matches `pattern`.
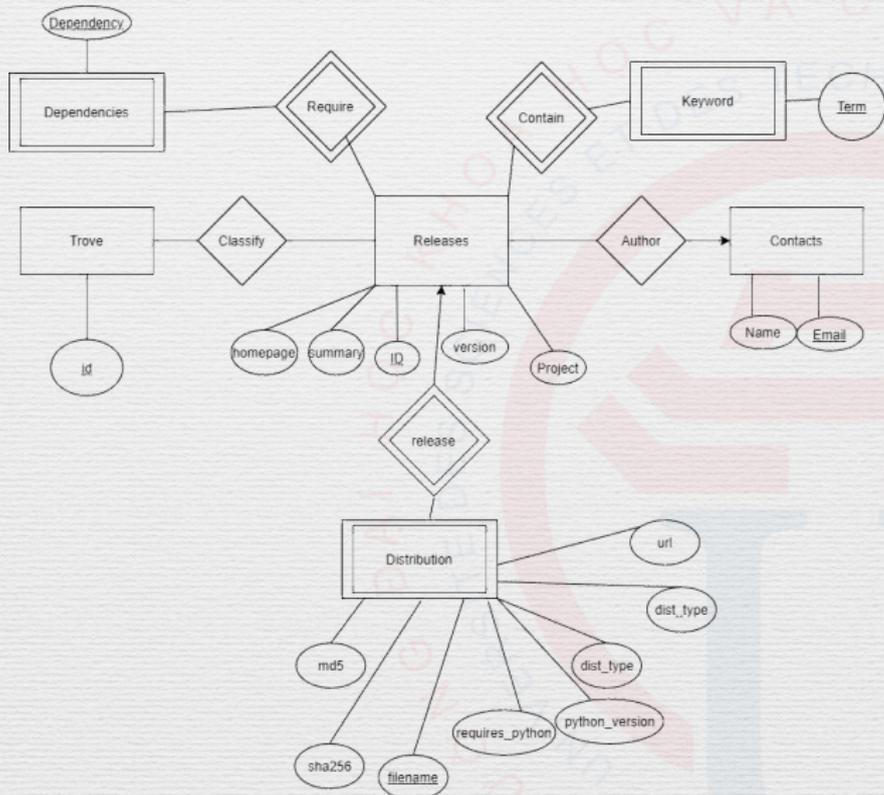
Python
Package
Metadata
Management

Group 8

Introduction
User
Requirements
Data
Definition
Entity Relationship
Diagram
Database Schema
Data Query
Conclusion

Database Schema

keywords
- release_id integer
- term varchar(32)

distributions
- release_id integer
- filename varchar(64)
- size integer
- url varchar(255)
- dist_type varchar(16)
- python_version varchar(8)
- requires_python varchar(32)
- sha256 varchar
- md5 varchar

releases
- id integer
- project varchar(32)
- version varchar(32)
- summary varchar(255)
- homepage varchar(2083)
- email varchar(255)

contacts
- email varchar(255)
- name varchar(255)

classifier
- release_id integer
- trove_id integer

troves
- id integer
- classifier varchar(255)

dependencies
- release_id integer
- dependency varchar(64)
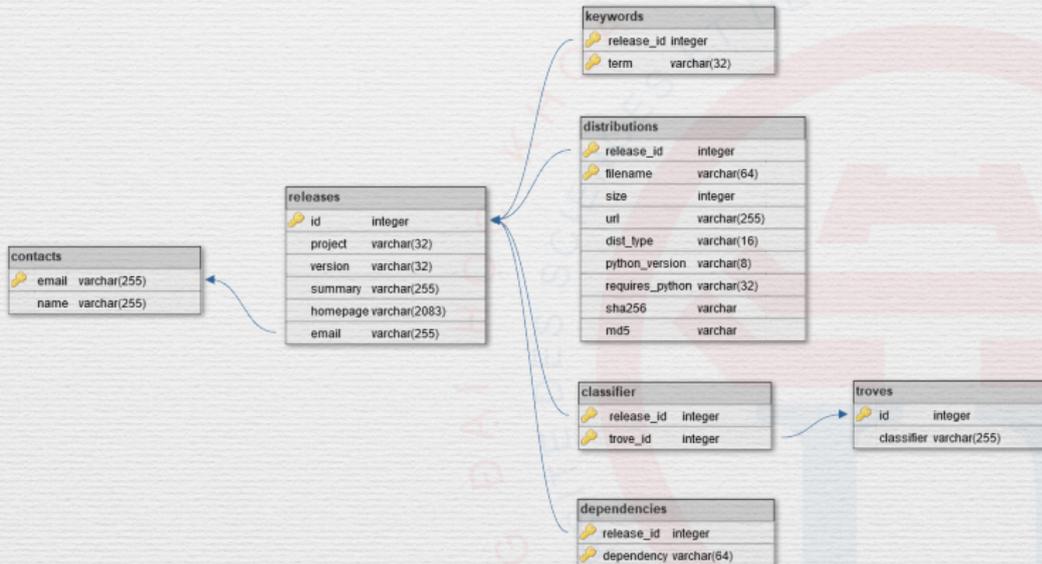
1 Introduction

2 User Requirements

3 Data Definition
   Entity Relationship Diagram
   Database Schema

4 Data Query

5 Conclusion

```
SELECT DISTINCT project
FROM releases;
```

```
SELECT version
FROM releases
WHERE project = 'spam'
ORDER BY version;
```

```
SELECT version
FROM releases
WHERE project = 'spam'
ORDER BY version DESC
LIMIT 1;
```

Project Listing by Author (view)

```
CREATE VIEW authorships
AS SELECT name as author, project
FROM contacts NATURAL JOIN releases
GROUP BY author, project;
```

Python
Package
Metadata
Management

Group 8

Project Listing by Author (query)

```
SELECT project
FROM authorships
WHERE author='Monty Python';
```

```
DELIMITER //
CREATE PROCEDURE browse(class varchar(255))
BEGIN
  SELECT project, version
  FROM releases, classifiers
  WHERE id = release_id AND trove_id = (
    SELECT id
    FROM troves
    WHERE classifier = class);
END//
DELIMITER ;
```

```
DELIMITER //
CREATE PROCEDURE release_data(
  project varchar(32), version varchar(32))
BEGIN
  DECLARE i smallint;
  SET i = (
    SELECT id
    FROM releases
    WHERE releases.project = project
      AND releases.version = version);
  SELECT project, version, homepage,
    name as author, email, summary
  FROM releases NATURAL JOIN contacts
  WHERE id = i;
```

Python
Package
Metadata
Management

Group 8

Release Metadata (cont.)

```
    SELECT term as keyword
    FROM keywords
    WHERE release_id = i;

    SELECT classifier
    FROM classifiers, troves
    WHERE release_id = i AND trove_id = troves.id;

    SELECT dependency
    FROM dependencies
    WHERE release_id = i;
END//
DELIMITER ;
```

```
SELECT project, version, summary
FROM releases
WHERE project LIKE 'py%';
```

Python
Package
Metadata
Management

Group 8

Introduction

User
Requirements

Data
Definition

Data Query

Conclusion

```
SELECT project, version, summary
FROM releases
WHERE summary LIKE '%num%';
```

❶ Introduction

❷ User Requirements

❸ Data Definition
    Entity Relationship Diagram
    Database Schema

❹ Data Query

❺ Conclusion

Python
Package
Metadata
Management

Group 8

What We Learnt

- Relational databases
- SQL—MySQL in particular
- Python package metadata format