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

Why?

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

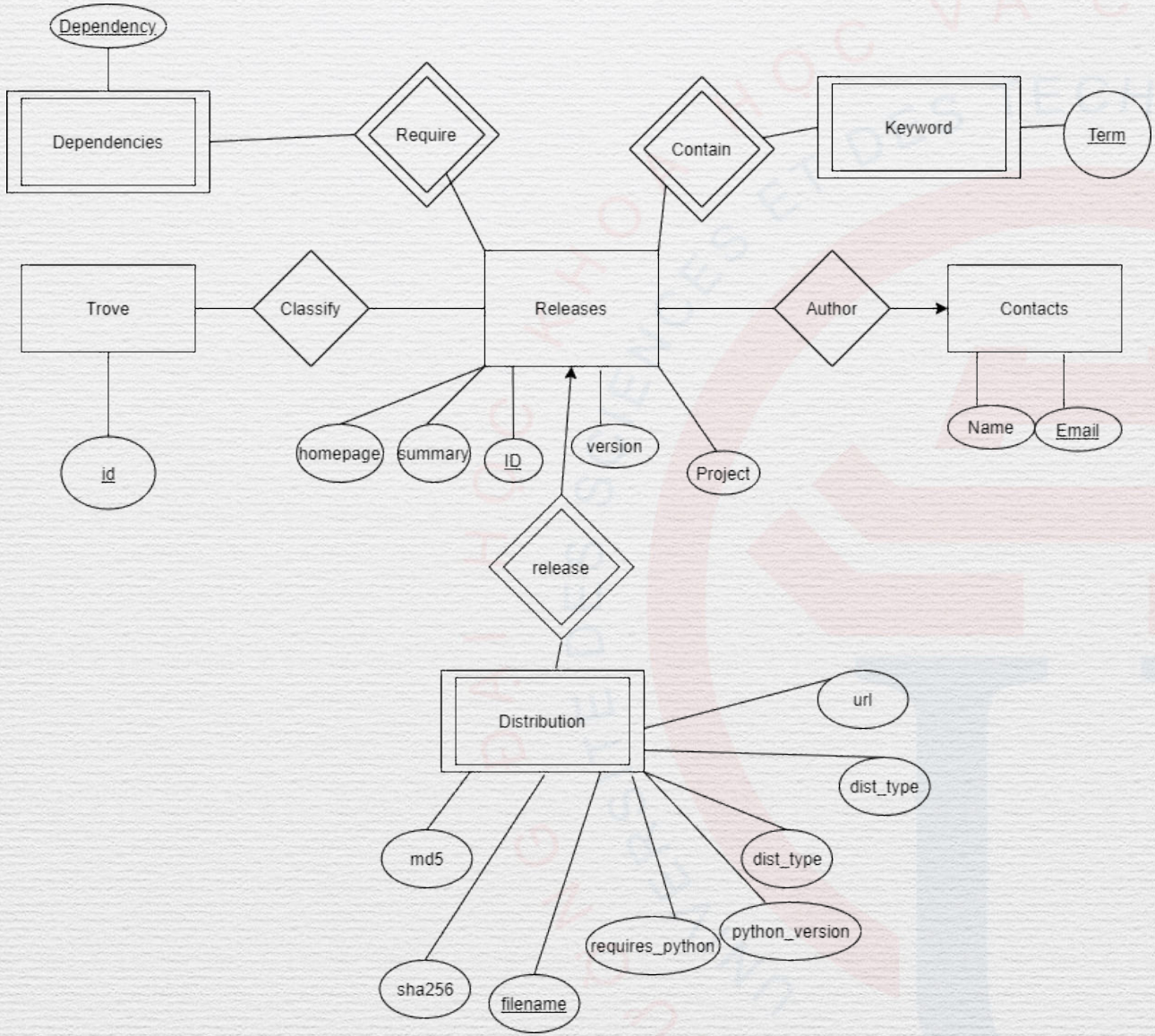
Tasks

- `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`.

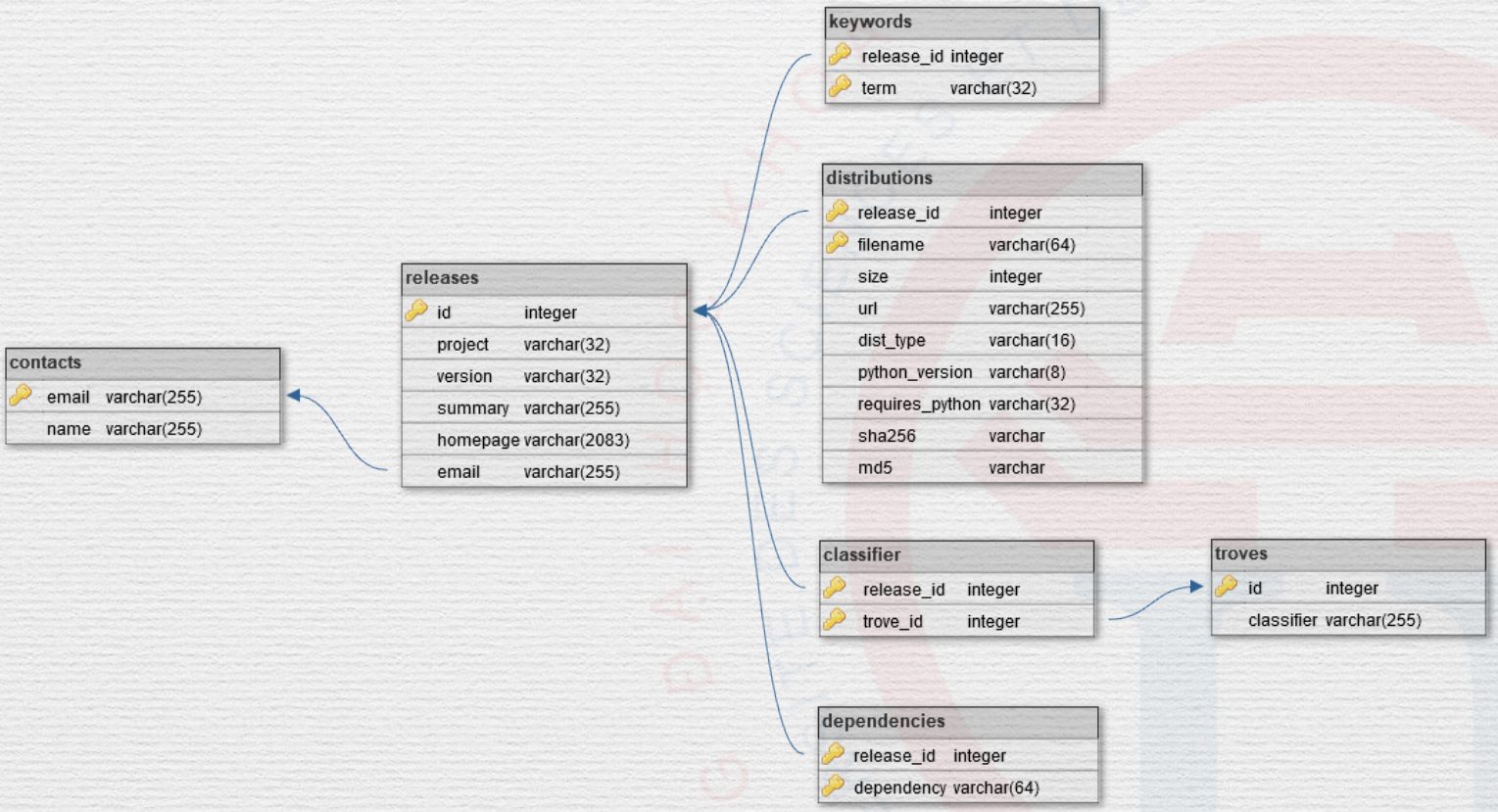
Tasks (cont.)

- `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`.

Entity Relationship Diagram



Database Schema



Project Listing

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

Project Release Listing

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

Project Latest Release

```
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;
```

Project Listing by Author (query)

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

Browse by Classifier

```
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 ;
```

Release Metadata

```
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;
```

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 ;
```

Project Search by Name

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

Project Search by Summary

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

What We Learnt

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

Copying