

Course ratings

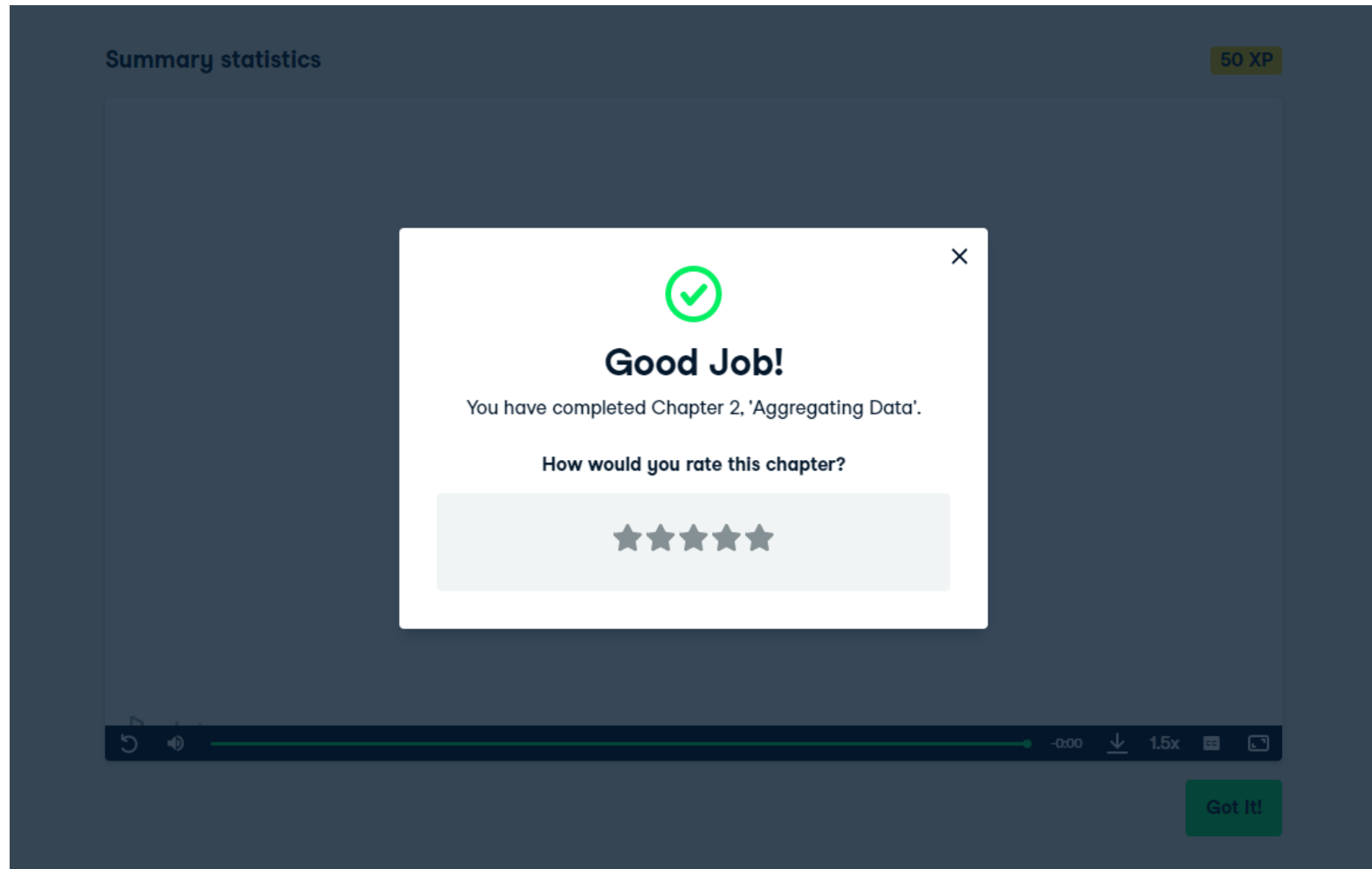
INTRODUCTION TO DATA ENGINEERING



Vincent Vankrunkelsven

Data Engineer @ DataCamp

Ratings at DataCamp

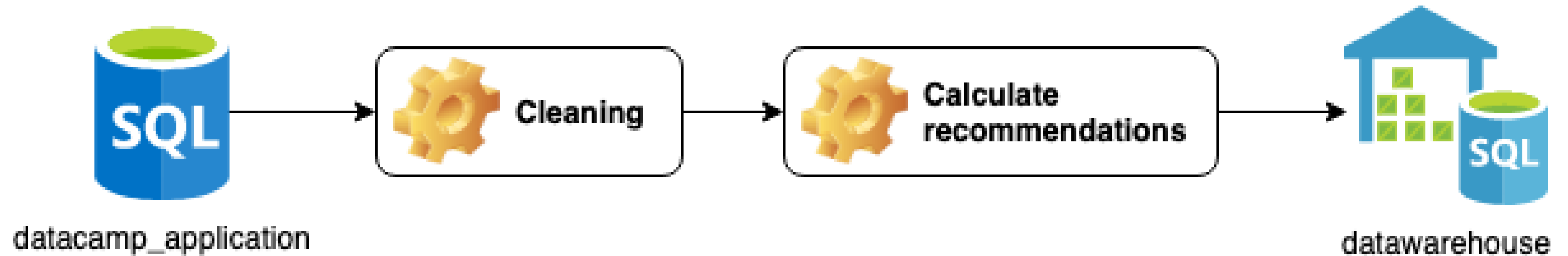


Recommend using ratings

- Get rating data
- Clean and calculate top-recommended courses
- Recalculate daily
- Example usage: user's dashboard

As an ETL process

It's an ETL process!

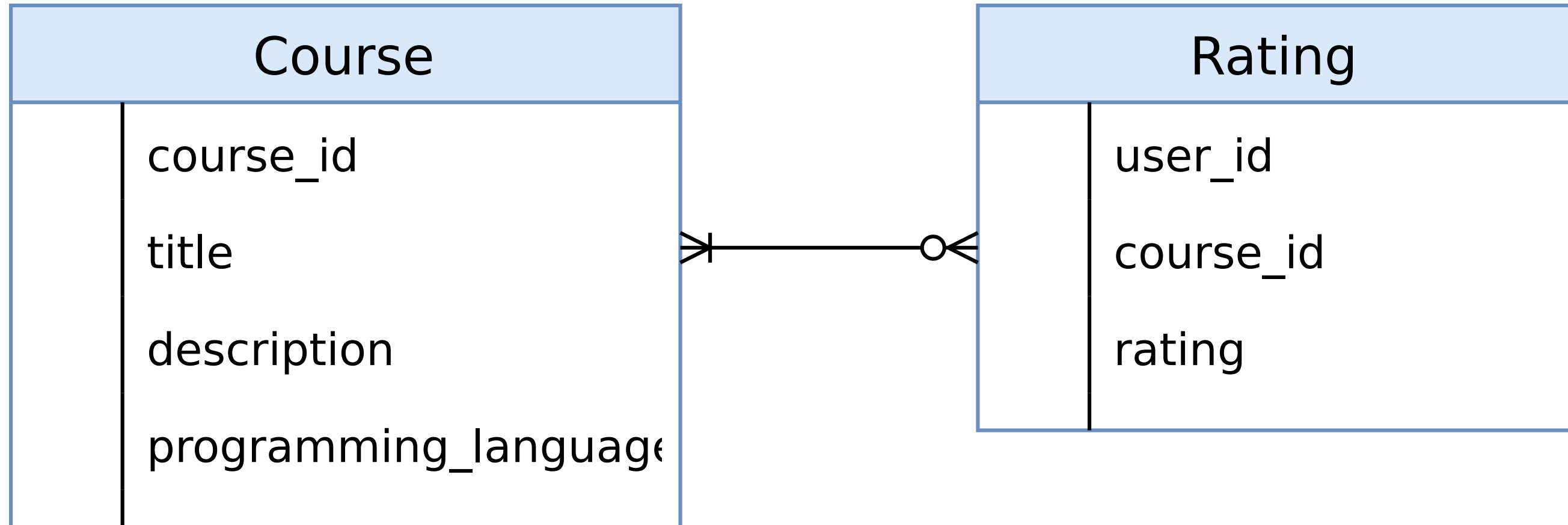


The database

| Course | |
|--------|----------------------|
| | course_id |
| | title |
| | description |
| | programming_language |

| Rating | |
|--------|-----------|
| | user_id |
| | course_id |
| | rating |

The database relationship



Let's practice!

INTRODUCTION TO DATA ENGINEERING

From ratings to recommendations

INTRODUCTION TO DATA ENGINEERING



Vincent Vankrunkelsven
Data Engineer @ DataCamp

The recommendations table

| user_id | course_id | rating |
|---------|-----------|--------|
| 1 | 1 | 4.8 |
| 1 | 74 | 4.78 |
| 1 | 21 | 4.5 |
| 2 | 32 | 4.9 |

The estimated rating of a course the user hasn't taken yet.

Recommendation techniques

- Matrix factorization
- Building Recommendation Engines with PySpark

Common sense transformation

| Course | |
|--------|----------------------|
| | course_id |
| | title |
| | description |
| | programming_language |

| Rating | |
|--------|-----------|
| | user_id |
| | course_id |
| | rating |

Recommendations

| user_id | course_id | rating |
|---------|-----------|--------|
| 1 | 1 | 4.8 |
| 1 | 74 | 4.78 |
| 1 | 21 | 4.5 |
| 2 | 32 | 4.9 |

Average course ratings

Average course rating

| course_id | avg_rating |
|-----------|------------|
| 1 | 4.8 |
| 74 | 4.78 |
| 21 | 4.5 |
| 32 | 4.9 |

We want to recommend highly rated courses

Use the right programming language

Rating

| user_id | course_id | programming_language | rating |
|---------|-----------|----------------------|--------|
| 1 | 1 | r | 4.8 |
| 1 | 74 | sql | 4.78 |
| 1 | 21 | sql | 4.5 |
| 1 | 32 | python | 4.9 |

Recommend SQL course for user with id 1

Recommend new courses

Rating

| user_id | course_id | programming_language | rating |
|---------|-----------|----------------------|--------|
| 1 | 1 | r | 4.8 |
| 1 | 74 | sql | 4.78 |
| 1 | 21 | sql | 4.5 |
| 1 | 32 | python | 4.9 |

Don't recommend the combinations already in the rating table

Our recommendation transformation

- Use technology that user has rated most
- Don't recommend courses that user already rated
- Recommend three highest rated courses from remaining combinations

Rating

| user_id | course_id | programming_language | rating |
|---------|-----------|----------------------|--------|
| 1 | 12 | sql | 4.78 |
| 1 | 52 | sql | 4.5 |
| 1 | 32 | r | 4.9 |

Recommend three highest rated SQL courses which are not 12 and 52.

Let's practice!

INTRODUCTION TO DATA ENGINEERING

Scheduling daily jobs

INTRODUCTION TO DATA ENGINEERING



Vincent Vankrunkelsven
Data Engineer @ DataCamp

What you've done so far

- Extract using `extract_course_data()` and `extract_rating_data()`
- Clean up using NA using `transform_fill_programming_language()`
- Average course ratings per course: `transform_avg_rating()`
- Get eligible user and course id pairs: `transform_courses_to_recommend()`
- Calculate the recommendations: `transform_recommendations()`

Loading to Postgres

- Use the calculations in data products
- Update daily
- Example use case: sending out e-mails with recommendations

The loading phase

```
recommendations.to_sql(  
    "recommendations",  
    db_engine,  
    if_exists="append",  
)
```

```
def etl(db_engines):  
    # Extract the data  
    courses = extract_course_data(db_engines)  
    rating = extract_rating_data(db_engines)  
    # Clean up courses data  
    courses = transform_fill_programming_language(courses)  
    # Get the average course ratings  
    avg_course_rating = transform_avg_rating(rating)  
    # Get eligible user and course id pairs  
    courses_to_recommend = transform_courses_to_recommend(  
        rating,  
        courses,  
    )  
    # Calculate the recommendations  
    recommendations = transform_recommendations(  
        avg_course_rating,  
        courses_to_recommend,  
    )  
    # Load the recommendations into the database  
    load_to_dwh(recommendations, db_engine)
```

Creating the DAG

```
from airflow.models import DAG
from airflow.operators.python_operator import PythonOperator

dag = DAG(dag_id="recommendations",
          scheduled_interval="0 0 * * *")
task_recommendations = PythonOperator(
    task_id="recommendations_task",
    python_callable=etl,
)
```

Let's practice!

INTRODUCTION TO DATA ENGINEERING

Congratulations

INTRODUCTION TO DATA ENGINEERING



Vincent Vankrunkelsven
Data Engineer @ DataCamp

Introduction to data engineering

- Identify the tasks of a data engineer
- What kind of tools they use
- Cloud service providers

Data engineering toolbox

- Databases
- Parallel computing & frameworks (Spark)
- Workflow scheduling with Airflow

Extract, Load and Transform (ETL)

- Extract: get data from several sources
- Transform: perform transformations using parallel computing
- Load: load data into target database

Case study: DataCamp

- Fetch data from multiple sources
- Transform to form recommendations
- Load into target database

Good job!

INTRODUCTION TO DATA ENGINEERING