SmartFridge

Alice Deng, Michael Deng, Barry Plunkett, Stephanie Shi

CIS 550

Basic Problems & Goals

1.	“What recipes can I make with **X** ingredients?”	Search by ingredient
2.	“What ingredients do I need to make **Y** recipe?”	Search by recipe
3.	“What recipes satisfy **Z** nutrition requirement ?”	Filter by nutrition

Preview

search by recipe search by ingredient smart fridge my cookbook

average calories	average protein	average sugar	average sodium	average cholesterol
6168.66	146.51 (g)	46.73 (g)	5637.69 (mg)	159.37 (mg)

clear cookbook

Tomato-Tortilla Soup

calories	4066.07
protein	271.04 (g)
sugar	21.33 (g)
sodium	14117.58 (mg)
cholesterol	430.12 (mg)

Poblano Posole

calories	12108.14
protein	84.36 (g)
sugar	49.67 (g)
sodium	879.50 (mg)
cholesterol	0.00 (mg)

Roasted Winter Vegetables

calories	2331.78
protein	84.13 (g)
sugar	69.18 (g)
sodium	1916.00 (mg)
cholesterol	48.00 (mg)

Datasets

We develop a web app enabling users to readily interpret information combined from **two** discrete data sources stored in **four** relations.

1. Eight Portions (JSON)

Recipes <> Ingredients
Recipes <> Pictures

Wrangling

- Conversion from JSON to CSV
- Locating pics from hashed urls
- Structured prediction model

Recipes
180,000 tuples

Images
80,000 t

2. USDA (CSV)

Ingredients <> Nutrition

Entity Resolution

- String processing on ingredient names
- Fuzzy string matching for ingredient matching

Ingredients
1.5 million t

USDA
9,000 t

More on Wrangling / Entity Resolution

Structured Prediction Model

- Allows storage of ingredients in normalized schema
- Extracts meaningful information from raw 'natural language ingredients'
  - Quantity, name, unit

Source: NYT Ingredient Phrase Tagger

String Processing

- On USDA ingredient names:
  - To lowercase
  - Remove dashes
  - Replace abbrevs / shorthands with full words

Fuzzy String Matching

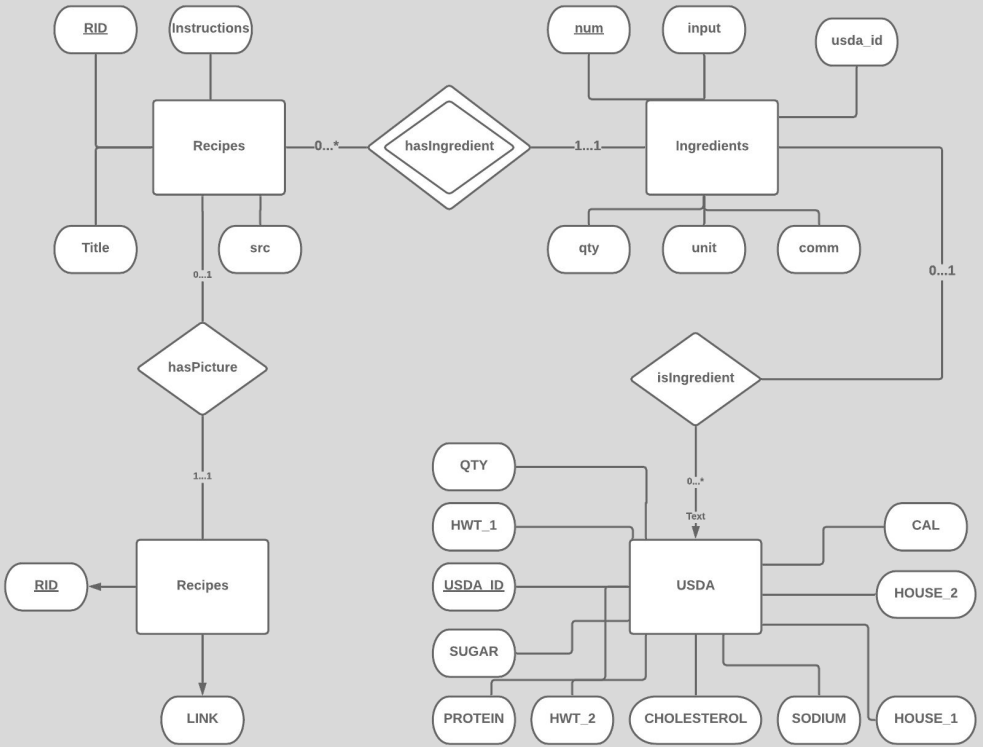
- For ingredient matching
- Various attempts
  - Levenshtein distance
  - **Normalized Levenshtein distance *(by length)***
  - Tokenized strings

Quantity Normalization

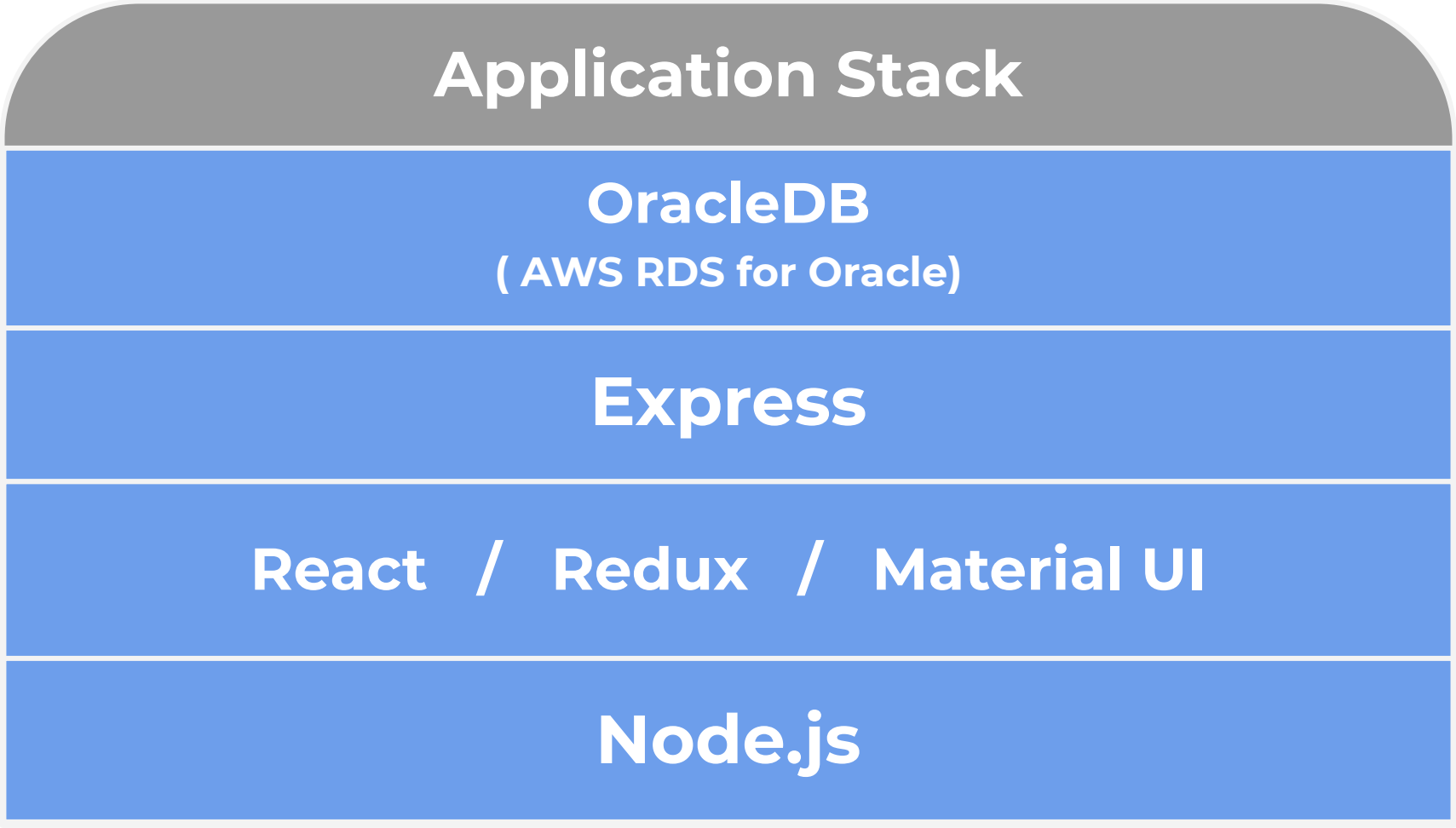
- For amount matching
- Examples:
  - teaspoon -> tsp
  - ounce -> oz
  - tablespoon -> tbsp

4-Relation Schema

Schema Design (3NF)



Demo


Application Stack
OracleDB
(AWS RDS for Oracle)
Express
React / Redux / Material UI
Node.js

Example: Complex Query

```
SELECT Q.TITLE AS TITLE, Q.RID AS RID, Z.LINK AS PICTURE_LINK FROM (SELECT
G.RID, G.TITLE
FROM
    (SELECT L.RID, L.TITLE
    FROM
        (SELECT RID, P.TITLE, T.TOTAL_COUNT, P.COUNT, P.COUNT /
T.TOTAL_COUNT AS SCORE
        FROM
        (SELECT RID, R.TITLE AS TITLE, COUNT(RID) AS COUNT
        FROM INGREDIENTS I NATURAL JOIN RECIPES R WHERE I.USDA_ID IN(16091,
1009, 19074, 18157)
        GROUP BY RID, R.TITLE) P
        NATURAL JOIN (SELECT RID, COUNT(*) AS TOTAL_COUNT FROM INGREDIENTS
GROUP BY RID) T
        ) L
    ORDER BY L.SCORE DESC) G
WHERE ROWNUM < 51) Q LEFT JOIN IMAGES Z ON Z.RID = Q.RID
```

Performance

Query Description	Original	Optimized
Get recipes by OR of ingredients	43 seconds	.02 seconds
Get recipes by AND of ingredients	6 seconds	.01 seconds
Get recipes by OR (sort by relevance)	59 seconds	.03 seconds
Get recipes by name (starts with)	12 seconds	.02 seconds
Get recipes by name (fuzzy)	Crashes	.3 seconds

Technical Challenges

1.	Data Cleaning & Entity Resolution	Use of third-party packages & machine learning
2.	Query Optimization	Creating indices, Browser-side filtering
3.	Fuzzy String Search	Reimplementation with caching

Thank You!

Questions?