



# Использование возможностей Oracle для вычисления персональных рекомендаций

---

Камиль Исламов, ООО «Предикт»

# План



- Простейшая задача персональных рекомендаций



- Выполнение расчётов с использованием MView



- Логика приложения через PL/SQL Package



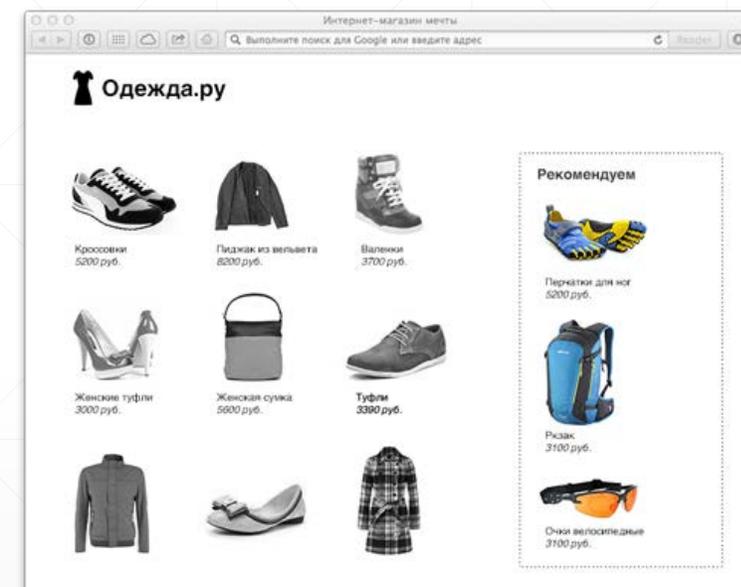
- Итоги



# Простейшая задача персональных рекомендаций



- Используется, например, в Интернет-магазинах
- Для вычисления рекомендаций достаточно
  - 2 пользователя
  - 2 действия
- Возможно решить без сложных математических вычислений
- Реализуется множеством технологий
- насыщает вебсайт дополнительным контентом
- Повышает SEO показатели



# Коллаборативная фильтрация



- Несколько методов вычисления «похожести»

$f_x$

- «Похожесть» вычисляется, например, по формуле косинусной меры

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|x\|_2 \times \|y\|_2}$$



- Исходные данные – поведенческие факторы посетителей



- Базовая единица данных – посещение пользователем какой-либо страницы

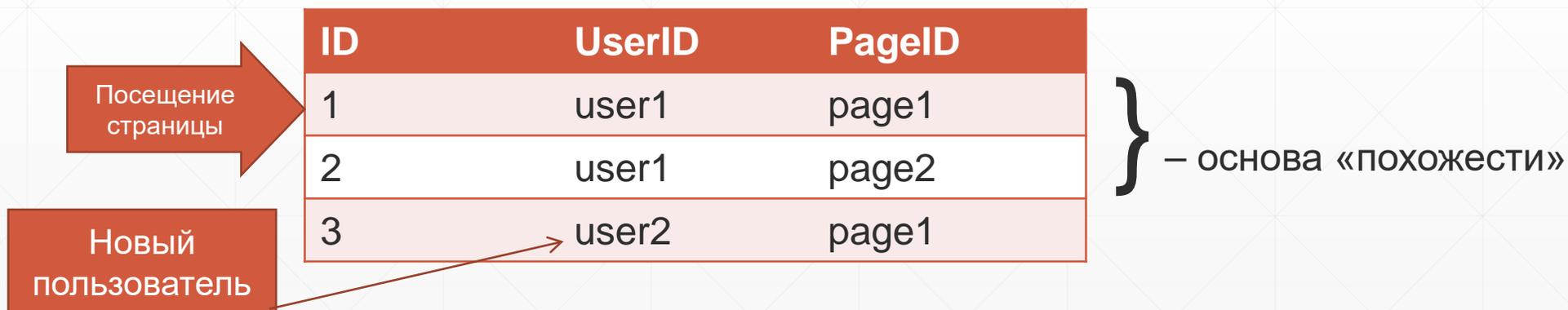
- User1 → Page1
- User1 → Page2
- User2 → Page1



# Структура данных поведенческих факторов

- Таблица «посещений» в БД: UserID1 → PageID1
- «Похожесть» для UserID2 вычисляется в момент посещения PageID\_
- Пересчитывается для UserID2 в реальном времени
- Расчёт полной базы «похожести» – периодически по требованию

Расчётная Модель



# Примеры вычисления похожести пользователей

UserID = 1

ID	PageID
1	page1
2	page2
3	page3

UserID = 2

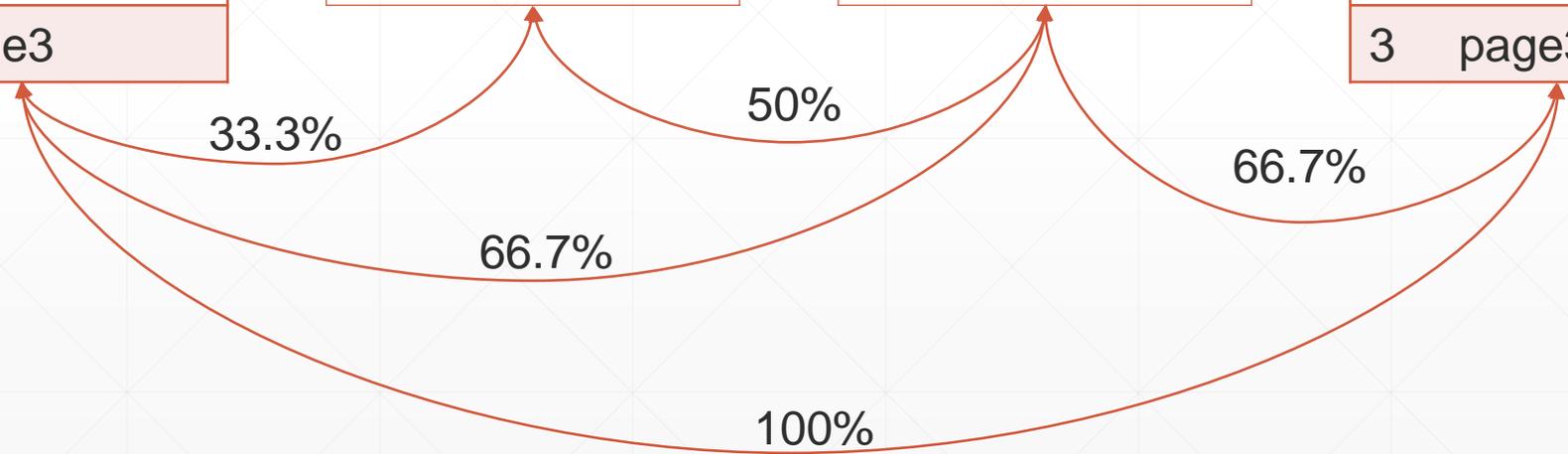
ID	PageID
1	page4
2	page2

UserID = 3

ID	PageID
1	page1
2	page2

UserID = 4

ID	PageID
1	page2
2	page1
3	page3



# Формула вычисления похожести пользователей

UserID = 1

ID	PageID
1	page1
2	page2
3	page3

UserID = 3

ID	PageID
1	page1
2	page2

Показатель	Значение
Количество совпадений	2
Общее количество	3
«Похожесть»	66.67%

UserID = 1

ID	PageID
1	page1
2	page2
3	page3

UserID = 4

ID	PageID
1	page2
2	page1
3	page3

Показатель	Значение
Количество совпадений	3
Общее количество	3
«Похожесть»	100%



# Особенности определения «похожести» элементов



- Предварительно вычисляется «похожесть» каждого элемента с каждым



- Для каждого элемента определяется список «похожих» элементов



- Список похожих элементов ранжируется по «похожести»



- Большое количество просмотров элементов увеличивает точность «похожести»



- Большое количество пользователей позволяет получить больше различных «похожестей»



# Хранение вычисленной «похожести» в БД

- Предварительно вычисленные коэффициенты item-to-item

Элемент 1	Элемент 2	Значение
Синий шарф	Красный шарф	0.73445
Синий шарф	Синяя сумочка	0.64323
Синий шарф	Жёлтый ботинок	0.0001
Белые туфли с каблуком	Белые туфли без каблука	0.6342
Белые туфли с каблуком	Белый шарф	0.53493
Белые туфли без каблука	Белый шарф	0.35233
Фиолетовое трико	Белый пиджак	0.0001



# Исходные данные для вычисления «похожести» элементов

- Данные для вычисления коэффициентов item-to-item «похожести»
  - 1 – посещение зафиксировано

	User1	User2	User3	User4	User5	User6
Item1	1	0	0	1	1	1
Item2	0	1	0	0	0	1
Item3	0	0	1	0	0	0
Item4	0	0	1	1	1	0
Item5	0	0	0	0	0	0
Item6	0	0	0	0	0	0



# Вычисление косинусной меры

- Вычисление коэффициента «похожести» Item1 ↔ Item2
  - Задействовано 2 строки

	User1	User2	User3	User4	User5	User6
Item1	1	0	0	1	1	1
Item2	0	1	0	0	0	1

- Количество разных пользователей Item1:  $c_1 = 4$
- Количество разных пользователей Item2:  $c_2 = 2$
- Количество пересекающихся пользователей:  $i = 1$
- Косинусная мера «похожести»  $x = \frac{i}{\sqrt{c_1 * c_2}} = \frac{1}{\sqrt{2 * 4}} = 0.3536$



# Вычисление косинусной меры в БД

Матрица исходных данных

	User1	User2	User3	User4	User5	User6
Item1	1	0	0	1	1	1
Item2	0	1	0	0	0	1
Item3	0	0	1	0	0	0
Item4	0	0	1	1	1	0
Item5	0	0	0	0	0	0
Item6	0	0	0	0	0	0

Хранение в БД

User	Item
1	1
2	2
3	3
3	4
4	1
4	4
5	1
5	4
6	1
6	2



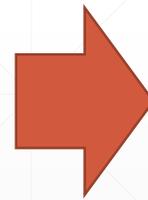
# Вычисление косинусной меры в БД

1. Объединение по Пользователю для разных Элементов

User	Item
1	1
2	2
3	3
3	4
4	1
4	4
5	1
5	4
6	1
6	2



User	Item
1	1
2	2
3	3
3	4
4	1
4	4
5	1
5	4
6	1
6	2



User	Item1	Item2
3	3	4
4	1	4
5	1	4
6	1	2



# Вычисление косинусной меры в БД

## 2. Подсчёт необходимых коэффициентов

User	Item	User	Item1	Item2	$c_1$	$c_2$	$i$	$x$
1	1	3	3	4	1	3	1	0.3333
2	2	4	1	4	4	3	2	0.5773
3	3	5	1	4	4	3	2	0.5773
3	4	6	1	2	4	2	1	0.3536
4	1							
4	4							
5	1							
5	4							
6	1							
6	2							

- Количество разных пользователей Item1:  $c_1$
- Количество разных пользователей Item2:  $c_2$
- Количество пересекающихся пользователей:  $i$
- Косинусная мера «похожести»  $x = \frac{i}{\sqrt{c_1 * c_2}}$



# Итоговые данные «похожести»



- На сколько один товар/материал «похож» на другой
- Для Item=1 здесь подходят Item=4 и Item=2



- Ранжирование для него: 4, 2

ID	Item1	Item2	Ratio
1	3	4	0.3333
2	1	4	0.5773
3	2	6	0.5773
4	1	2	0.3536



# Реализация в Oracle



- Таблица *Событий* – просмотр UserID какого-либо ItemID (`user2item`)



- Материализованные представления вычисления «похожести»



- Отдельный *Mview* для каждого алгоритма



- Выдача всех алгоритмов рекомендаций через единую функцию



- Динамический SQL для переключения между MView



# Materialized View Косинусной меры

```
SELECT DISTINCT item1, item2, ratio
FROM (SELECT u1.user1,
            u1.item1,
            u2.item2,
            COUNT (DISTINCT u1.user1)
              OVER (PARTITION BY u1.item1, u2.item2)
            / SQRT ( COUNT (DISTINCT item2) OVER ()
                  * COUNT (DISTINCT item1) OVER ()) ratio
FROM user2item u1, user2item u2
WHERE u1.user1 = u2.user2 AND u2.item2 <> u1.item1)
```

$$\frac{i}{\sqrt{c_1 * c_2}}$$

- Количество пересекающихся пользователей:  $i$
- Количество разных пользователей Item1:  $c_1$
- Количество разных пользователей Item2:  $c_2$



# Единая PLSQL функция выдачи рекомендаций



- Mview по алгоритму косинусной меры COSINE\_MV



- Mview по алгоритму Танимото TANIMOTO\_MV



- Параметры функции:

- Item1
- Алгоритм
- Количество



- Динамический SQL для переключения между Mview

```
SELECT item2  
FROM (SELECT item2  
FROM <ALGORYTHM>_MV where item1=:item1  
order by ratio DESC) where rownum <= :COUNT
```



# Единый Раскаге персональных рекомендаций



- Функция сбора событий `user2item`



- Функция актуализации всех Mview `cosine_mv`



- Функция получения рекомендаций по разным алгоритмам



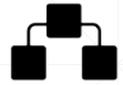
- Другие вспомогательные функции



# Логика приложения вне Oracle



- Сервер приложений обеспечивает запуск функций пакета



- Планируется система маршрутов



- Запускается соответствующий маршруту метод из Package

- В каждом Методе выполняется одна транзакция с одной функцией



- Один POST/GET запрос – один вызов соответствующей функции



# Возможности Oracle для персонализации



- Эффективная разработка бизнес-логики через Package



- Массовая загрузка данных используя bulk функционал



- Эффективные JOIN для таблиц больших объёмов



- Удобные материализованные представления



- Оперативное обновление MView



- В совокупности – это *мощная платформа* для быстрого запуска



# Преимущества Oracle для персонализации



- Все преимущества реализации бизнес-логики через Package



- Полноценное использование возможностей Oracle



- Единое пространство для реализации всех задач



- Отсутствие операций обмена данными на внешние системы



- Экономия ресурсов на разработку при оптимальной эффективности



