Introduction to data driven decision making

DATA-DRIVEN DECISION MAKING IN SQL

SQL

Irene OrtnerData Scientist at Applied Statistics



Aim of this course

- A short review of SQL know-how
- Apply your SQL know-how to extract business insights from data
- Learn about new SQL statements to summarize data
 - OLAP extensions were developed specifically for business intelligence
 - Examples are CUBE, ROLLUP and GROUPING SETS

MovieNow: an online movie rental company

- Platform to stream movies
- Additional information for each movie: genre, main actors, etc.
- Customer information
- Customers can give a rating after watching a movie

customer_id
name
country
gender
date_of_birth
date_account_start

customers

customer_id
name
country
gender
date_of_birth
date_account_start

movies

movie_id
title
genre
runtime
year_of_releas
renting_price



customers

customer_id
name
country
gender
date_of_birth
date_account_start

movies

movie_id
title
genre
runtime
year_of_releas
renting_price

renting

renting_id customer_id movie_id rating date_renting



customers

customer_id
name
country
gender
date_of_birth
date_account_start

movies

movie_id
title
genre
runtime
year_of_releas
renting_price

renting

renting_id customer_id movie_id rating date_renting actors

actor_id name year_of_birth nationality gender

customers

customer_id
name
country
gender
date_of_birth
date_account_start

movies

movie_id
title
genre
runtime
year_of_releas
renting_price

renting

renting_id customer_id movie_id rating date_renting actors

actor_id name year_of_birth nationality gender actsin

actsin_id movie_id actor_id

Objectives of data driven decision making

- Information for operational decisions
 - Popularity of actors to decide which movies to invest in.
 - Revenue of the last months to estimate budget for short term investments.
- Information for strategic decisions
 - Success across countries to decide on market extensions.
 - Longterm development of revenue for long term investments.

KPIs: Key Performance Indicators

Extract information from the data which is relevant to measure the success of MovieNow.

- Total number of rentals: revenue
- The average rating of all movies: customer satisfaction
- Number of active customers: customer engagement

Let's get started!

DATA-DRIVEN DECISION MAKING IN SQL



Filtering and ordering

DATA-DRIVEN DECISION MAKING IN SQL



Tim Verdonck

Professor Statistics and Data Science



WHERE

Select all customers from Italy:

```
SELECT *
FROM customers
WHERE country = 'Italy';
```

```
|customer_id | name
                                | country | gender | date_of_birth | date_account_start|
                                                    1999-07-21
 53
            | Teresio Panicucci | Italy
                                                                    2018-11-06
                                          male
              Demetrio Palermo
                                                   1997-10-10
                                                                    2018-10-17
 54
                                | Italy
                                          male
 55
            | Facino Milano
                                | Italy
                                                   1973-05-23
                                                                   2018-01-02
                                          male
```

Operators in the WHERE clause

- Comparison operators:
 - Equal =
 - Not equal <>
 - Less than <
 - Less than or equal to <=
 - Greater than >
 - Greater than or equal to >=
- BETWEEN operator
- IN operator
- IS NULL and IS NOT NULL operators

Example comparison operators

Select all columns from movies where the genre is not Drama.

```
SELECT *
FROM movies
WHERE genre <> 'Drama';
```

Select all columns from movies where the price for renting is larger equal 2.

```
SELECT *
FROM movies
WHERE renting_price >= 2;
```

Example: BETWEEN operator

Select all columns of customers where the date when the account was created is between 2018-01-01 and 2018-08-31.

```
SELECT *
FROM customers
WHERE date_account_start BETWEEN '2018-01-01' AND '2018-09-31';
```

Example: IN operator

Select all actors with nationality USA or Australia.

```
SELECT *
FROM actors
WHERE nationality IN ('USA', 'Australia')
```



Example: NULL operator

Select all columns from renting where rating is NULL.

```
SELECT *
FROM renting
WHERE rating IS NULL
```

Select all columns from renting where rating is not NULL.

```
SELECT *
FROM renting
WHERE rating IS NOT NULL
```

Boolean operators AND

Select customer name and the date when they created their account for customers who are from Italy AND who created an account between 2018-01-01 and 2018-08-31.

```
SELECT name, date_account_start
FROM customers
WHERE country = 'Italy'
AND date_account_start BETWEEN '2018-01-01' AND '2018-08-31';
```

Boolean operators OR

Select customer name and the date when they created their account for customers who are from Italy _OR_ who created an account between 2018-01-01 and 2018-08-31.

```
SELECT name, date_account_start
FROM customers
WHERE country = 'Italy'
OR date_account_start BETWEEN '2018-01-01' AND '2018-08-31';
```

ORDER BY

Order the results of a query by rating.

```
SELECT *
FROM renting
WHERE rating IS NOT NULL
ORDER BY rating;
```

renting_id	customer_id	movie_id	rating	date_renting
552	28		1	 2017-03-27
558	41	19	3	2019-01-13
444	120	59	3	2018-08-10
200	86	46	3	2018-08-26
234	104	28	4	2018-10-04

ORDER BY ... DESC

Order the results of a query by rating in descending order.

```
SELECT *
FROM renting
WHERE rating IS NOT NULL
ORDER BY rating DESC;
```

renting_id	customer_id	movie_id	rating	date_renting
243	7	 5	10	 2019-01-11
18	36	39	10	2019-03-20
396	7	40	10	2018-09-11
487	61	48	10	2017-08-14
476	78	42	10	2018-07-04

Let's practice!

DATA-DRIVEN DECISION MAKING IN SQL



Aggregations - summarizing data

DATA-DRIVEN DECISION MAKING IN SQL



Bart Baesens

Professor Data Science and Analytics



Overview aggregations

```
SELECT AVG(renting_price)
FROM movies;
```



Overview aggregations

```
SELECT AVG(renting_price)
FROM movies;
```

Some aggregate functions in SQL

- AVG()
- SUM()
- COUNT()
- MIN()
- MAX()

Aggregation with NULL values

```
SELECT COUNT(*)
FROM actors;
```

• Result: 145

```
SELECT COUNT(name)
FROM actors;
```

• Result: 145

```
SELECT COUNT(year_of_birth)
FROM actors;
```

• Result: 143

DISTINCT

```
SELECT DISTINCT country
FROM customers;
```

```
SELECT COUNT(DISTINCT country)
FROM customers;
```

• Result: 11

DISTINCT with 'NULL' values

```
SELECT DISTINCT rating
FROM renting
ORDER BY rating;
```

Give an alias to column names

Helps to understand the results when column names are self-explaining.

Let's practice!

DATA-DRIVEN DECISION MAKING IN SQL

