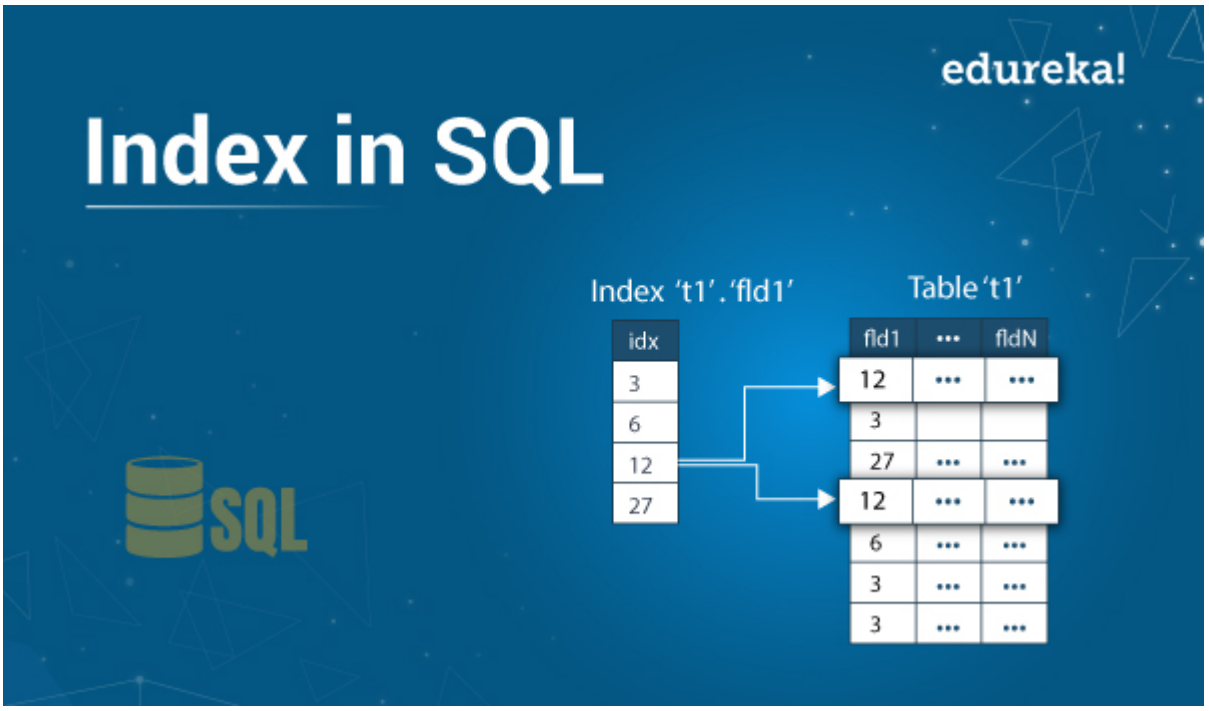
P9. Work In pairs. Index Importance



Tobal Moll **Adrián Piña**

INDEX

PART 1

EXERCISE

Load the TENNIS.sql and TENNIS_NOPK_NOFK.sql . They are the same database but the second one has no primary keys and no foreign keys. As a consequence, the second has no indexes. Run the query you will find below in both databases and note how long it takes to produce a result.

```
SELECT CONCAT_WS(' ', P1.p_name, P1.p_surname) AS Player1,
CONCAT_WS(' ', P2.p_name, P2.p_surname) AS Player2, t_name AS Tournament,
t_end_date AS `End date`, IF (mr_winner = R1.r_num, CONCAT_WS(' ', 'Winner:',
P1.p_name, P1.p_surname), CONCAT_WS(' ', 'Winner:', P2.p_name, P2.p_surname)) AS
Winner
FROM REGISTRATIONS R1, REGISTRATIONS R2, MATCHES, MATCH_RESULTS,
TOURNAMENTS, PLAYERS P1, PLAYERS P2
WHERE
m_id = mr_m_id AND
  ((R1.r_num = m_r_num1 AND R2.r_num = m_r_num2) OR
    (R1.r_num = m_r_num2 AND R2.r_num = m_r_num1) ) AND
       t_id = m_t_id AND
       t_num_rounds = m_round AND
       t_type = 'Singles' AND
       R2.r_p_id = P2.p_id AND
       R1.r_p_id = P1.p_id AND
       P1.p_id < P2.p_id;
```

Explain, as we saw in the Socratica video, and detail your conclusions regarding the existence (or not) of the indexes in the previous database.

SOLUTION

The conclusions that we have drawn from the video is that the index is a two-column table that relates tables or views, this is done to improve the efficiency of searches in the databases of the elements that may be important or necessary, such as example foreign keys or primary keys.

Next we will show the difference between having a database with indexes and one that does not have indexes. For this we will use the databases and the query that were provided to us previously.

TENNIS.sql

```
TENNIS> SELECT CONCAT_WS(' ', P1.p_name, P1.p_surname) AS Player1,
        CONCAT_WS(' ', P2.p_name, P2.p_surname) AS Player2, t_name AS Tournament, t_end_date AS `End dat
        FROM REGISTRATIONS R1, REGISTRATIONS R2, MATCHES, MATCH_RESULTS, TOURNAMENTS, PLAYERS P1, PLAYER
        WHERE
        m_id = mr_m_id AND
          ((R1.r_num = m_r_num1 AND R2.r_num = m_r_num2) OR
            (R1.r_num = m_r_num2 AND R2.r_num = m_r_num1) ) AND
                t_id = m_t_id AND
                t_num_rounds = m_round AND
                t_type = 'Singles' AND
                R2.r_p_id = P2.p_id AND
                R1.r_p_id = P1.p_id AND
                P1.p_id < P2.p_id
[2021-04-13 20:08:58] 8 rows retrieved starting from 1 in 124 ms (execution: 86 ms, fetching: 38 ms)
```

	Player1	Player2	Tournament	`End date`	Winner
1	Roger Federer	Rafael Nadal	French Open	2007-06-10	Winner: Rafael Nadal
2	Roger Federer	Rafael Nadal	Wimbledon	2007-07-08	Winner: Roger Federer
3	Roger Federer	Fernando Gonzalez	Australian Open	2007-01-28	Winner: Roger Federer
4	Roger Federer	Novak Djokovic	US Open	2007-09-09	Winner: Roger Federer
5	Tommy Robredo	David Ferrer	Heineken Open	2007-01-14	Winner: David Ferrer
6	James Blake	Radek Stepanek	Countrywide Classic	2007-07-22	Winner: Radek Stepanek
7	Mikhail Youzhny	Philipp Kohlschreiber	BMW Open	2007-05-06	Winner: Philipp Kohlschreiber
8	Juan Carlos Ferrero	Guillermo Canas	Brasil Open 2007	2007-02-18	Winner: Guillermo Canas

TENNIS_NOPK_NOFK.sql

```
TENNIS2> SELECT CONCAT_WS(' ', P1.p_name, P1.p_surname) AS Player1,
        CONCAT_WS(' ', P2.p_name, P2.p_surname) AS Player2, t_name AS Tournament, t_end_date AS `End date`, IF (mr_wi
        FROM REGISTRATIONS R1, REGISTRATIONS R2, MATCHES, MATCH_RESULTS, TOURNAMENTS, PLAYERS P1, PLAYERS P2
        WHERE
        m_id = mr_m_id AND
          ((R1.r_num = m_r_num1 AND R2.r_num = m_r_num2) OR
            (R1.r_num = m_r_num2 AND R2.r_num = m_r_num1) ) AND
                t_id = m_t_id AND
                t_num_rounds = m_round AND
                t_type = 'Singles' AND
                R2.r_p_id = P2.p_id AND
                R1.r_p_id = P1.p_id AND
                P1.p_id < P2.p_id
[2021-04-13 20:16:00] 8 rows retrieved starting from 1 in 5 m 9 s 642 ms (execution: 5 m 9 s 624 ms, fetching: 18 ms)
```

	Player1	Player2	Tournament	`End date`	Winner
1	Roger Federer	Rafael Nadal	French Open	2007-06-10	Winner: Rafael Nadal
2	Roger Federer	Rafael Nadal	Wimbledon	2007-07-08	Winner: Roger Federer
3	Roger Federer	Novak Djokovic	US Open	2007-09-09	Winner: Roger Federer
4	Mikhail Youzhny	Philipp Kohlschreiber	BMW Open	2007-05-06	Winner: Philipp Kohlschreiber
5	Tommy Robredo	David Ferrer	Heineken Open	2007-01-14	Winner: David Ferrer
6	Juan Carlos Ferrero	Guillermo Canas	Brasil Open 2007	2007-02-18	Winner: Guillermo Canas
7	James Blake	Radek Stepanek	Countrywide Classic	2007-07-22	Winner: Radek Stepanek
8	Roger Federer	Fernando Gonzalez	Australian Open	2007-01-28	Winner: Roger Federer

As we can see, the difference in execution time between one table and another is quite large. In the table that does not have indexes, the query takes 5 minutes to give us the results, while the query that has indexes takes 1 second.

PART 2

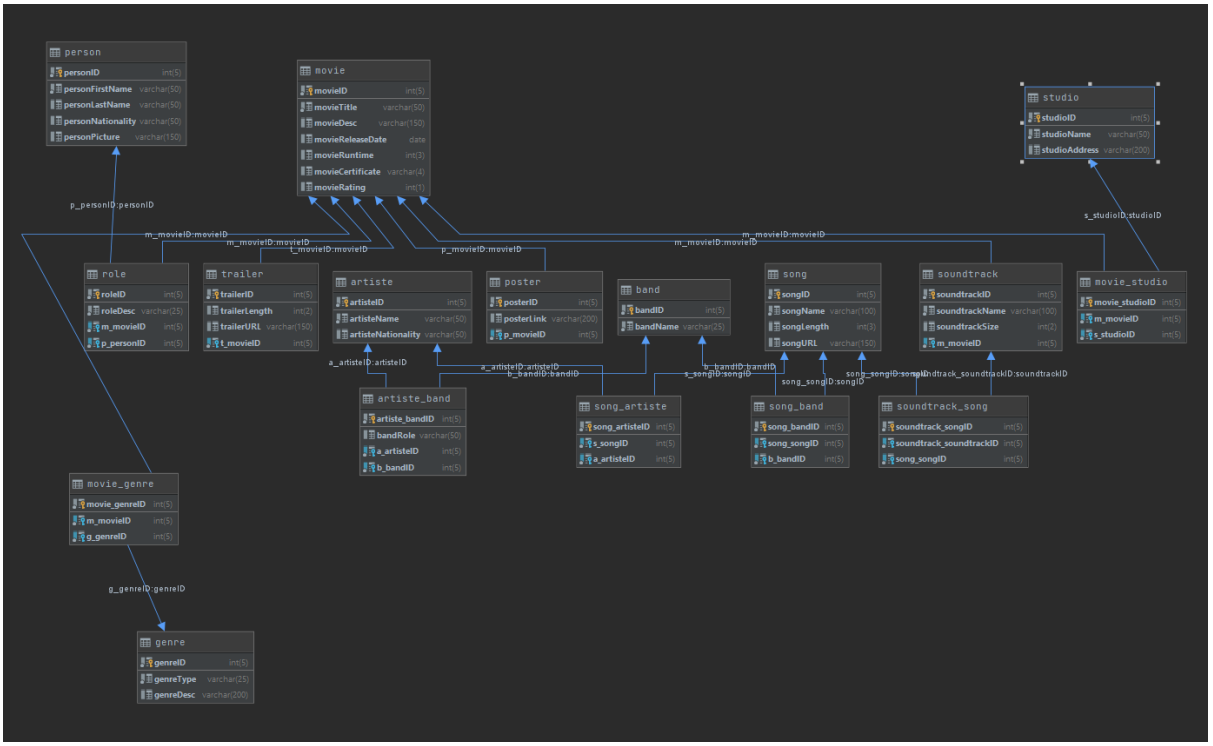
EXERCISE

Search in the internet for a database, preferably from an open source, including indexes (and not including it, in order to compare its behaviour) in any of its tables and implement some advanced queries like in the previous part.

SOLUTION

In this exercise we have decided to create two databases, one has indexes and primary keys while the other does not. We have done this to be able to compare the behavior of the two tables and thus be able to see how much the indexes make our work easier.

The database that we are going to use is called [MoviesDB](#).



What are indexes?

Indexes allow you to locate and return records in a simple and fast way. They are especially useful when we want to search for elements among the millions of records that a table can contain at any given time. When we do not use indexes, we can sometimes perceive that it takes too long to answer a query.

Test

The first test that we are going to perform will be a basic query. In the following images we can see that in the table with indexes it takes 141 ms while the table that does not have indexes takes 144 ms, we can see that the difference is almost non-existent and even the database without indexes is faster.

with indexes

```
sequelmovie> select * from artiste
[2021-04-13 23:08:04] 32 rows retrieved starting from 1 in 78 ms (execution: 66 ms, fetching: 12 ms)
```

```
sequelmovie> select * from artiste, band,song
[2021-04-13 23:07:02] 500 rows retrieved starting from 1 in 141 ms (execution: 97 ms, fetching: 44 ms)
```

without indexes

```
sequelmovie2> select * from artiste
[2021-04-13 23:07:49] 32 rows retrieved starting from 1 in 78 ms (execution: 65 ms, fetching: 13 ms)
```

```
sequelmovie2> select * from artiste, band,song
[2021-04-13 23:07:07] 500 rows retrieved starting from 1 in 144 ms (execution: 98 ms, fetching: 46 ms)
```

For this second test we will see what happens with a more complex query, in the following images, the data base with indexes takes 103 ms while the data base that has no indexes takes 14 seconds. We can see that the time difference is quite large between one and the other.

with indexes

```
[2021-04-13 22:14:53] 125 rows retrieved starting from 1 in 103 ms (execution: 91 ms, fetching: 12 ms)
```

without indexes

```
[2021-04-13 22:14:07] 125 rows retrieved starting from 1 in 14 s 802 ms (execution: 14 s 789 ms, fetching: 13 ms)
```

Conclusion

As we have seen in basic queries, the indices do not give us a notable improvement in efficiency, but on the other hand, in complex queries we do have quite a notable improvement that will help us save time and work efficiently.