Index

Tamas Piros

Purpose of an index

- Data is stored on disk as blocks of data
- Accessing this data requires linear search (requires block access)
  - linear search means the bigger the list of items to search the more time it will take to execute
- Indexing creates another data structure which holds the field values only and pointer to the record it relates to
- This index structure is sorted allowing for binary searches
  - binary search starts in the middle of the sorted list and eliminates parts of the list as candidates

Index

id	name	age
1	John	18
2	Adam	34
3	Simon	22
4	Christina	28

- Linear search to look for Adam against the table:
  - Is Adam at Position 1? No
  - Is Adam at Position 2? Yes, match.
  - Is Adam at Position 3? No.
  - Is Adam at Position 4? No.
  - More Items? No. Stop search.

value	Pointer
Adam	2
Christina	4
John	1
Simon	3

- Binary search to look for Adam against the index:
  - Is Adam in the first half? Yes
  - Is Adam in the first position? Yes, match.
  - Does Adam appear more times in the first half? No. Stop search

Create an index

- At table creation
  - `CREATE TABLE t(…, INDEX (column1));`
- Later point in time
  - `CREATE INDEX [name] ON table (column1);`
- Normally creates a BTREE structure

Additional commands to be familiar with

- `DROP INDEX [name];`
- `SHOW INDEX FROM [table];`
- All indexes

    -
      ```
      USE information_schema;
      SELECT * FROM statistics;
      ```

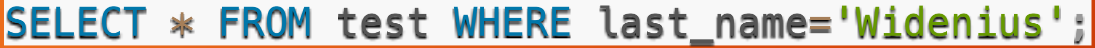
Index types

- Single column (most common, allows for fast lookup)
- Multiple column (“composite” indexes) - in MySQL up to 16 columns
  - Sometimes called ‘compound’ index
- FULLTEXT (for text search)
- Spatial (geometry, distances, geographic shapes)

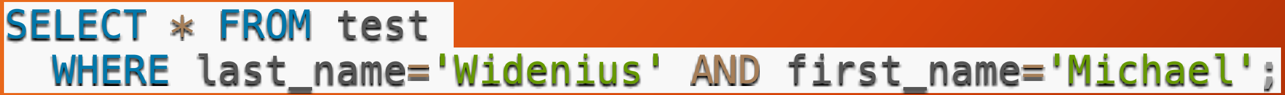
Composite Index

```
CREATE TABLE test (
    id          INT NOT NULL,
    last_name   CHAR(30) NOT NULL,
    first_name  CHAR(30) NOT NULL,
    PRIMARY KEY (id),
    INDEX name (last_name,first_name)
);
```

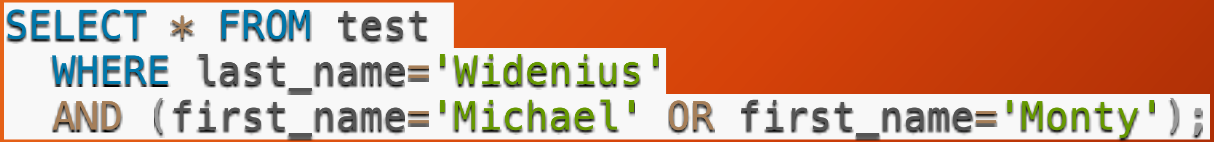
Composite Index



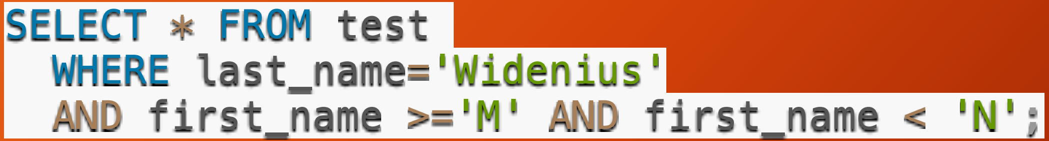
```
SELECT * FROM test WHERE last_name='Widenius';
```



```
SELECT * FROM test
  WHERE last_name='Widenius' AND first_name='Michael';
```



```
SELECT * FROM test
  WHERE last_name='Widenius'
  AND (first_name='Michael' OR first_name='Monty');
```



```
SELECT * FROM test
  WHERE last_name='Widenius'
  AND first_name >='M' AND first_name < 'N';
```