

INDEX

MATA KULIAH PEMROGRAMAN BASIS DATA
TAHUN AJARAN 2023/2024

arranged by Novi Prisma Yunita, M.Kom



- Index digunakan untuk menemukan baris dengan nilai kolom yang spesifik dengan cepat
- Tanpa index, pencarian baris dilakukan ke seluruh baris pada tabel, sehingga semakin besar isi tabel semakin besar juga biaya yang dibutuhkan untuk mencapai pencarian tertentu
- Dengan index, MySQL dapat dengan cepat menemukan baris tanpa harus mencari ke seluruh baris data tanpa harus mencari secara sekuensial

- Untuk menemukan baris yang sesuai dengan klausa WHERE
- Untuk menghilangkan baris yang tidak dipertimbangkan
- Untuk mengambil baris dari tabel lain dengan JOIN
- Untuk menemukan nilai MIN () dan MAX() dari kolom yang terindeks dengan index key tertentu

Hindari Penggunaan Index

- Pada tabel dengan jumlah baris kecil (kurang dari 10 baris, dengan nilai kolom yang cenderung pendek)
- Pada tabel besar tetapi query-nya cenderung menampilkan seluruh isi tabel. Saat query membutuhkan result-set besar, membaca baris secara sekuensial lebih cepat ketimbang menggunakan index karena pencarian sekuensial dapat meminimalisir pencarian dalam disk

Composite Index

- Composite index disebut juga multiple column index adalah index dengan beberapa kolom sebagai index key
- Jumlah kolom maksimal pada composite index adalah 16 kolom
- Jika index dengan composite key tidak menyebutkan urutan index dari kolom pada index key pertama, maka index tidak akan bekerja pada query tersebut

Composite Index

- Jika tabel memiliki *multiple column index* (index dengan index key berasal dari beberapa kolom), maka index key yang paling kiri akan digunakan sebagai optimizer untuk mencari baris.
- Misalnya: index dengan 3 buah kolom (col1, col2, col3) maka pencarian bisa dilakukan dengan: (col1), (col1, col2), dan (col1, col2, col3)

Composite Index

Index key-nya adalah col1, col2, dan col3. Mana saja dari query di bawah ini yang index-nya bekerja?

SELECT * FROM tbl_name WHERE col1=val1;

SELECT * FROM tbl_name WHERE col1=val1 AND col2=val2;

SELECT * FROM tbl_name WHERE col2=val2;

SELECT * FROM tbl_name WHERE col2=val2 AND col3=val3;

```
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)  
);
```




Mana dari query berikut yang bekerja menggunakan index?

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

```
SELECT * FROM test
```

```
WHERE last_name='Jones' OR first_name='John';
```

```
SELECT * FROM test WHERE last_name='Jones' AND first_name='John';
```

```
SELECT * FROM test WHERE first_name='John';
```

```
SELECT * FROM test
```

```
WHERE last_name='Jones' AND (first_name='John' OR first_name='Jon');
```

```
SELECT * FROM test
```

```
WHERE last_name='Jones' AND first_name >='M' AND first_name < 'N';
```

Statement for Index

- **CREATE**
- **SHOW**
- **DROP**

CREATE INDEX

```
CREATE [UNIQUE] INDEX index_name  
ON table_name (columnName [ASC|DESC] [, ...])
```

Penjelasan

- **UNIQUE:** UNIQUE bersifat opsional, penambahan klausa UNIQUE pada saat pembuatan index akan memaksakan nilai unik pada kolom yang didaftarkan sebagai *index key*. Sebaliknya, tanpa penambahan UNIQUE, *index key* dibolehkan untuk memiliki nilai duplikat. UNIQUE hanya bisa diterapkan terhadap kolom yang memiliki *constraint* UNIQUE/PRIMARY KEY
- **index_name:** nama index
- **table_name:** nama tabel yang akan dibuat *index*-nya.

Contoh (1)

```
CREATE UNIQUE INDEX idx_customer_id  
ON customers (customer_id);
```

Contoh (2)

```
CREATE UNIQUE INDEX idx_item_id  
ON items (item_id DESC);
```


Contoh (3) membuat *index* saat creating table.

```
CREATE TABLE t (  
    t_id INT PRIMARY KEY,  
    c1 INT,  
    c2 INT,  
    INDEX(c1) ) ;
```

Contoh (4) Membuat index dengan ALTER TABLE

```
ALTER TABLE t ADD INDEX (c2) ;
```



SHOW LIST OF INDEX

Untuk menampilkan index dari tabel tertentu:

```
SHOW INDEXES FROM table_name;
```

Untuk menampilkan index dari tabel di basisdata tertentu:

```
SHOW INDEXES FROM table_name IN database_name;
```

atau,

```
SHOW INDEXES FROM database_name.table_name;
```

Perintah di atas akan menampilkan *result-set* yang sama

DROP INDEX



```
DROP INDEX index_name ON table_name;
```

```
DROP INDEX c1 ON t;
```



QUESTION ?