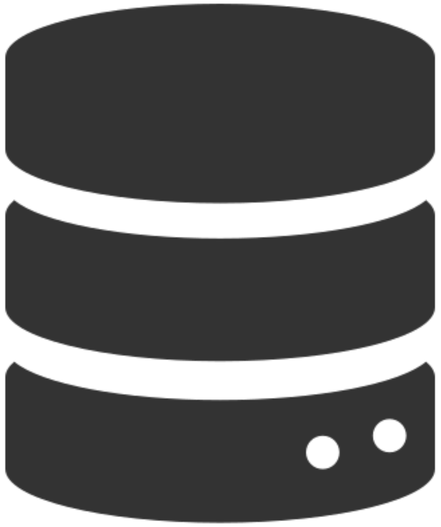
인덱스 (Index)

김정욱

2024.08.01

Index

인덱스란?



인덱스 (Index)

데이터베이스 분야에 있어서
테이블에 대한 동작의 속도를
높여주는 자료 구조를 일컫는다.
(B Tree)

출처: 위키백과

Index

인덱스란?

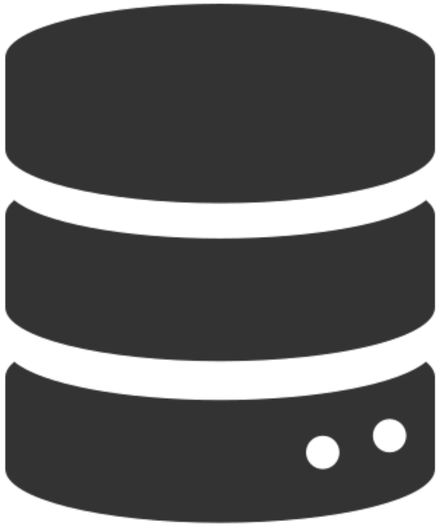
Query

company_id	pointer
10	_126
11	_131
12	_132
13	_137

pointer	company_id	units	unit_cost
_126	10	12	1.15
_131	11	5	2.32
_132	12	6	5.32
_137	13	43	2.32
_140	14	32	1.42
_141	15	13	2.12

Index

인덱스 특징



인덱스 (Index)

데이터들이 **오름차순으로 정렬**

출처: 위키백과

Index

인덱스 장점

테이블을 조회하는 속도와
그에 따른 성능을 향상

Index

인덱스 장점

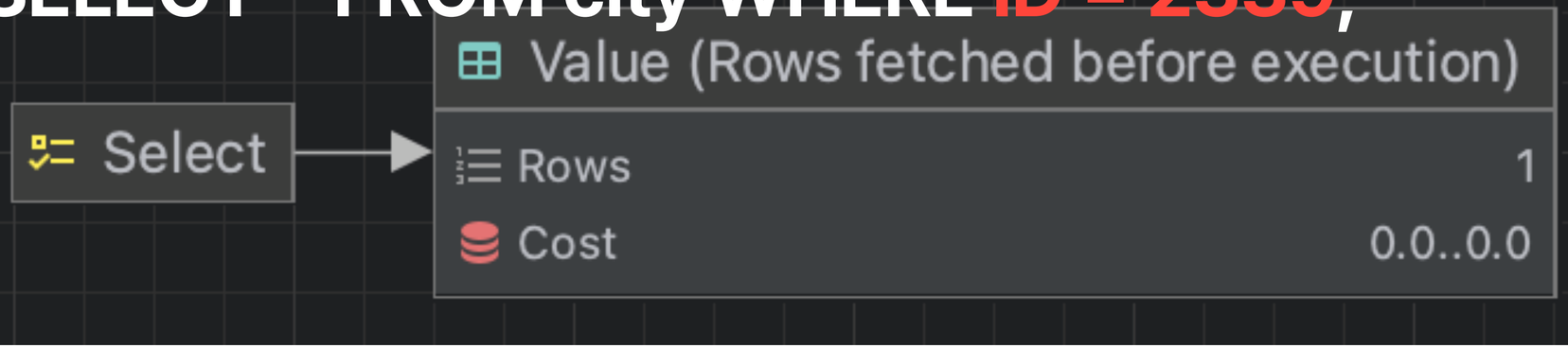
1. 조건 검색 WHERE절의 효율성
2. 정렬 OREDR BY절의 효율성
3. MIN, MAX의 효율적인 처리가 가능

출처: [DB] 데이터베이스(DB) 인덱스(Index)란 무엇인가?

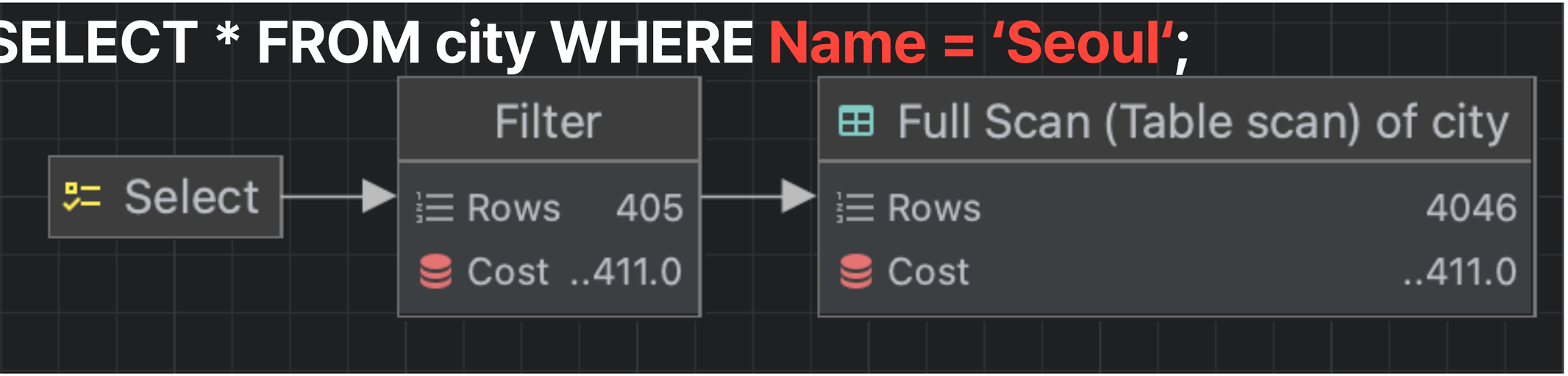
Index

인덱스 장점 - WHERE 절

```
SELECT * FROM city WHERE ID = 2339;
```




```
SELECT * FROM city WHERE Name = 'Seoul';
```



Index

인덱스 장점 - WHERE 절

SELECT * FROM city WHERE ID = 2339;

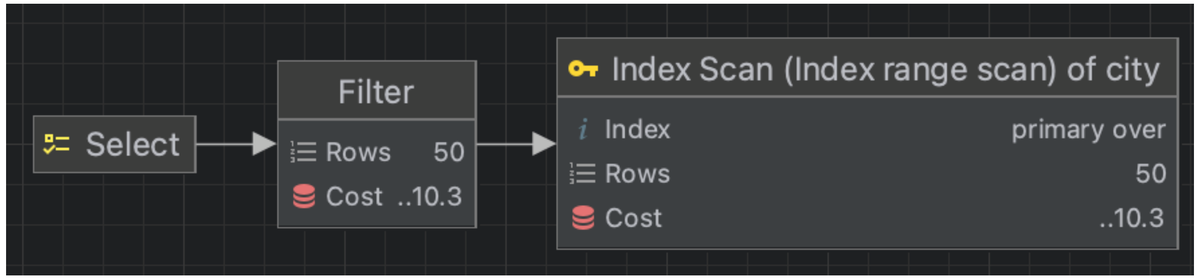
Rows: 1
Cost: 0.0

SELECT * FROM city WHERE Name = ‘Seoul’;

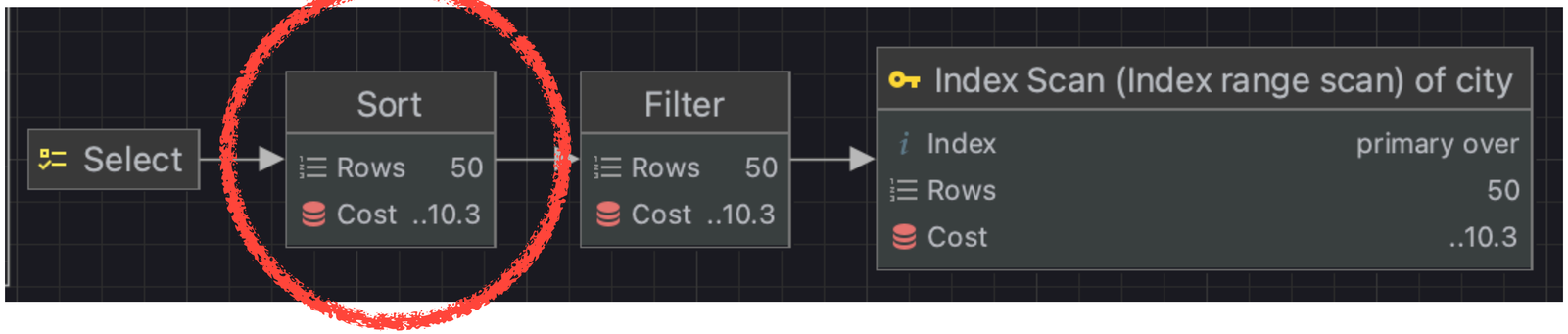
Rows: 4046
Cost: 411.0(Filter) + 411.0(Table Scan) = 822.0

Index

인덱스 장점 - ORDER BY 절




```
SELECT *
FROM city
WHERE ID > 50 AND ID <= 100
OREDER BY ID desc;
```




```
SELECT *
FROM city
WHERE ID > 50 AND ID <= 100
OREDER BY Name desc;
```

Index

인덱스 장점

테이블을 조회하는 속도와
그에 따른 성능을 향상

전반적인 시스템의 부하를 줄임

Index

인덱스 단점

1. 인덱스를 관리하기 위해 DB의
   약 10%에 해당하는 저장공간 필요
2. 인덱스를 관리하기 위한 추가 작업 필요
3. 잘못 사용할 경우 오히려 성능이 저하(DML)

출처: [DB] 데이터베이스(DB) 인덱스(Index)란 무엇인가?

Index

인덱스 단점

빈번한 테이블 수정(DML)

인덱스 테이블과 원본 테이블 수정 작업

Index

인덱스 관리

1. INSERT: 새로운 데이터에 대한 인덱스 추가
2. DELETE: 삭제하는 인덱스를 사용하지 않는다는 작업 진행
3. UPDATE: 기존의 인덱스를 사용하지 않음 처리하고, 갱신된 데이터에 대해 인덱스 추가

오버헤드 발생

출처: [DataBase] 인덱스(index)란?

Index

Index의 종류

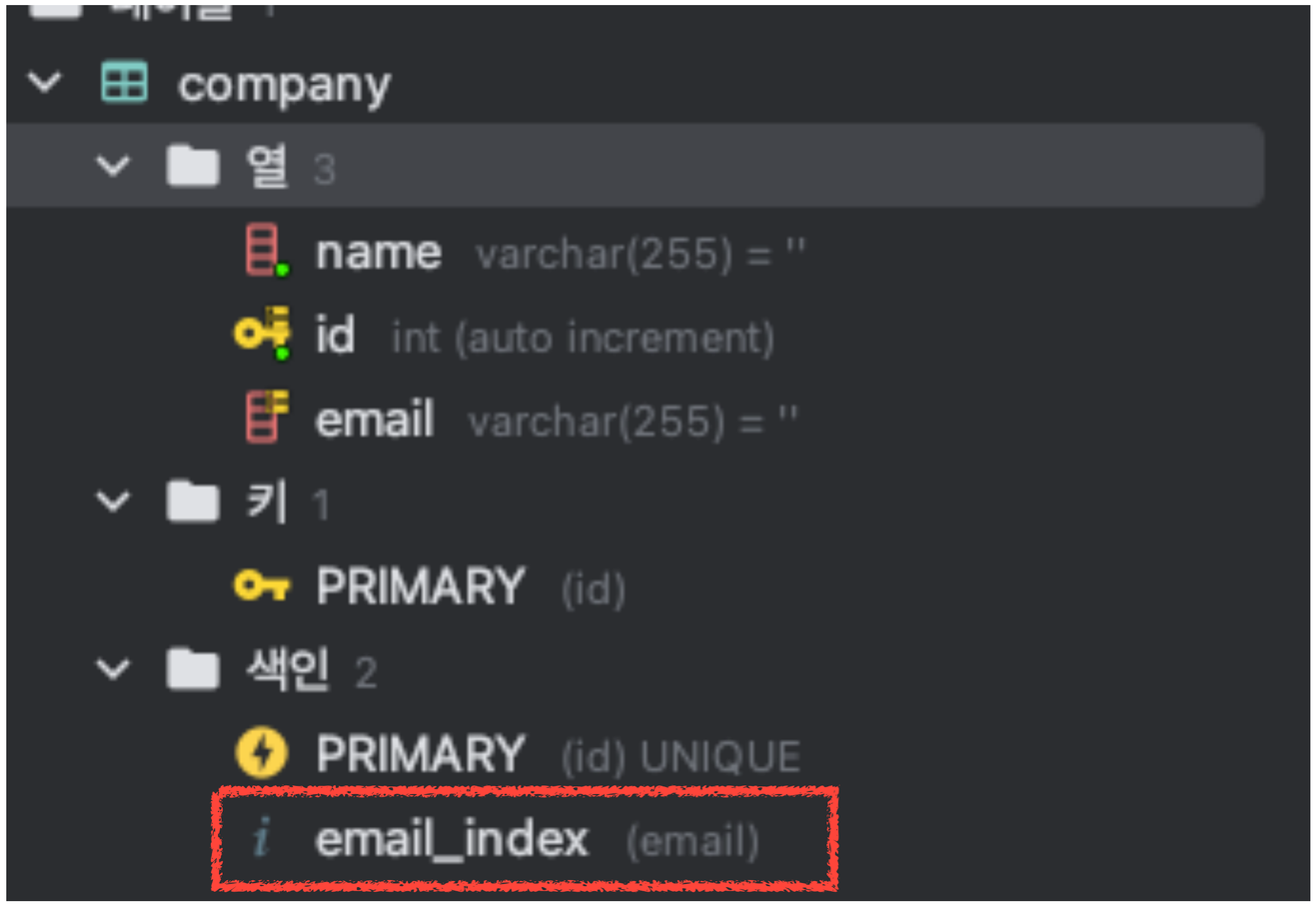
1. 인덱스(Index)
2. 고유 인덱스(Unique Index)
3. 고유 제약 조건 (Unique Constraint)

출처: [DB] UNIQUE INDEX 사용 이유

Index

Index의 종류 - Index

CREATE INDEX index_name ON tbl_name (column)



Index

Index의 종류 - Unique Index

열의 제약 조건에 Unique 가있으면
mongoose가 자동으로 인덱스를 생성

Primary Key도 Unique 조건이
기본으로 있으니 적용됨

Index

Index의 종류 - Unique Constraint

Unique Constraint는
Unique Index로 구현

Index

Index의 종류 - Unique Constraint

그러면 왜 있을까?

Index

Index의 종류 - Unique Constraint

목적을 명시하는 목적

Index

Index의 종류 - Unique Constraint

1. 인덱스 기능을 중점적으로 활용하여 고유성을 강조
2. 비즈니스 규칙을 중점적으로 강조

출처: [DB] UNIQUE INDEX 사용 이유

Index

어떠한 경우에 Index를 사용하는가?

1. 카디널리티 (Cardinality) 👍
   중복도가 낮으면 좋다

2. 선택도 (Selectivity) 👎
   한 컬럼이 갖고 있는 값 하나로 적은 row를 찾는다

3. 조회 활용도 👍

4. 수정 빈도 👎

출처: 효과적인 DB index 설정하기

Index

어떠한 경우에 Index를 사용하는가?

1. WHERE에 자주 사용되는 컬럼
2. LIKE와 사용할 경우에는 %가 뒤에 사용되도록 하기
3. ORDER BY에 자주 사용되는 컬럼
4. JOIN에 자주 사용되는 컬럼

출처: 효과적인 DB index 설정하기

Index

PK는 Index인가?

PK는 자동으로 인덱스를 생성하지만
물리적으로 저장되지는 않는다!

Index

FK는 Index인가?

DB마다 다르고, MySQL같은 경우에는 생성

InnoDB

Index

복합 인덱스

데이터베이스에서 여러 개의 컬럼(열)들을
조합하여 인덱스를 생성하는 것

Index

복합 인덱스

```
CREATE INDEX index_name
ON tbl_name (column1, column2)
             A        B
```

1. A열을 기준으로 인덱스를 생성후
2. B열 기준으로 인덱스를 생성

Index

복합 인덱스

SELECT * FROM city WHERE A = 123

Index

복합 인덱스

```
SELECT * FROM city
WHERE A = 123 AND B = 'Seoul'
```

Index

복합 인덱스

```
SELECT * FROM city
WHERE B = 'Seoul'
```

Index

복합 인덱스 장점

1. 검색 속도 개선
2. 데이터 정렬의 효율성
3. 인덱스의 용량 절감
4. 쿼리 최적화

출처: 복합인덱스 (Composite Index)

Index

복합 인덱스 주의점

1. 인덱스를 생성하는 열의 개수가 많아질수록 성능은 떨어짐
2. 인덱스 생성 순서도 고려
3. WHERE 절에서 먼저 사용되는 컬럼을 앞쪽에 위치시키는 것이 좋음

출처: 복합인덱스 (Composite Index)

Index

Q & A

Reference

- [DB] 데이터베이스(DB) 인덱스(Index)란 무엇인가?
- [Database] 인덱스(index)란?
- 위키백과 - 인덱스 (데이터베이스)
- [DB] key와 index의 차이
- 효과적인 DB index 설정하기