SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20031

Computing Technology Design Project

Week 09

Indexes

Database Development Lifecycle

1. Planning
2. Requirement gathering
3. Conceptual design
4. Logical design
5. Physical design
6. **Construction**
7. **Implementation** & rollout
8. Ongoing support

Outline

1. What & Why index?
2. Creating an index
3. Showing indexes
4. Dropping indexes
5. Multicolumn indexes
    a. performance experiment
6. Project update

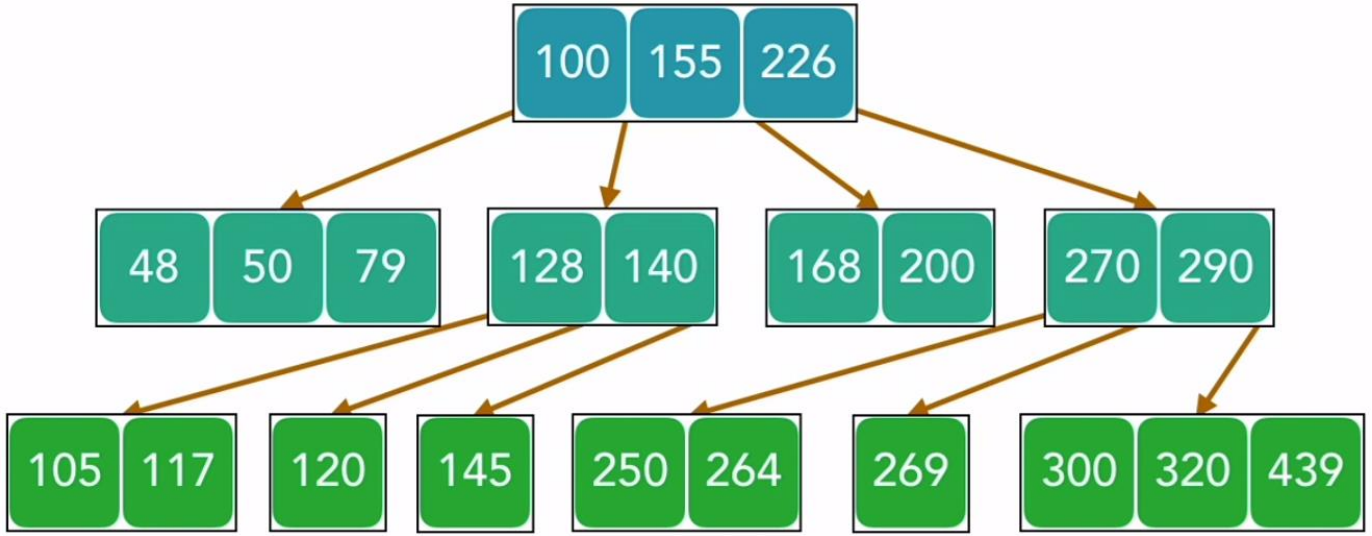
(A) What & Why indexes

What is index?

- Index is a data structure that helps (esp. in large data set)
  - rapidly look up data (WHERE clause)
  - enforce constraints
  - sort data (ORDER BY clause)
- Costs associated with index:
  - performance cost: needs updating when data is updated (insert, update, drop)
  - storage cost
  - should only be used with selected columns
- Common index design: B-Tree (Binary tree)
- Indexes are dropped when the associated table is

Binary Tree (B-Tree)

Find 145?



When to use index

- Primary key
- Columns used for ORDER BY
- Columns used in WHERE clauses:
  - col = value
  - col > value

(B) Creating an index

Creating an index

- Add to the CREATE TABLE statement
  - can be UNIQUE or Non-UNIQUE

```
CREATE TABLE test (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    string1 VARCHAR(128),
    string2 VARCHAR(128),
    INDEX str1(string1)
);
```

implicit index on PK (also unique)

explicit index on column (non-unique)

```
SHOW INDEX FROM test;
```

unique-ness

index name

data structure

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
test	0	PRIMARY	1	id	A	0	*NULL*	*NULL*		BTREE
test	1	str1	1	string1	A	0	*NULL*	*NULL*	YES	BTREE

```
CREATE TABLE test (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    string1 VARCHAR(128) UNIQUE,
    string2 VARCHAR(128),
    UNIQUE INDEX str1(string1)
);
SHOW INDEX FROM test;
```

UNIQUE indexes on column

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
test	0	PRIMARY	1	id	A	0	*NULL*	*NULL*		BTREE
test	0	string1	1	string1	A	0	*NULL*	*NULL*	YES	BTREE
test	0	str1	1	string1	A	0	*NULL*	*NULL*	YES	BTREE

Creating an index (2)

- Use CREATE INDEX statement (after table creation)
  - can be UNIQUE or Non-UNIQUE

```
CREATE INDEX str2 ON test(string2);

OR

CREATE UNIQUE INDEX str2 ON test(string2);


SHOW INDEX FROM test;
```

(C) Showing indexes

Showing indexes (1): in one table

- Use SHOW INDEX FROM statement
- Variations:
  - SHOW INDEXES FROM…
  - SHOW INDEXES IN …

Showing indexes (1): in all tables in a schema

```
SELECT DISTINCT table_name, index_name
FROM information_schema.statistics
  WHERE table_schema = 'scratch';
```

table_name	index_name
customer	PRIMARY
item	PRIMARY
numerics	PRIMARY
sale	PRIMARY
test	PRIMARY
test	str2
test	str1

(D) Dropping indexes

Dropping indexes

```
DROP INDEX str1 ON test;

SHOW INDEX FROM test;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
test	0	PRIMARY	1	id	A	0	*NULL*	*NULL*		BTREE
test	0	str2	1	string2	A	0	*NULL*	*NULL*	YES	BTREE

(D) Multicolumn indexes

Multicolumn indexes

- Index on more than one columns

```
CREATE TABLE test (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    string1 VARCHAR(128),
    string2 VARCHAR(128),
    INDEX twostrs (string1,string2)
);
```

can also add **UNIQUE**

Multicolumn indexes

```
SHOW INDEX FROM test;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name
test	0	PRIMARY	1	id
test	1	twostrs	1	string1
test	1	twostrs	2	string2

Multicolumn indexes: explanation

```
EXPLAIN SELECT string1, string2
    FROM test
    ORDER BY string1, string2;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	test	index	*NULL*	twostrs	774	*NULL*	1	Using index

Single column index: explanation

```
EXPLAIN SELECT string1
    FROM test
    ORDER BY string1;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	test	index	*NULL*	str1	387	*NULL*	1	Using index

Experiment: performance impact of index

- Drop table test
- Recreate it with the 2 column index
- Insert 45-50 records into the table
- Case 1: query with index (record exec. time)
  - use the 2-column SELECT query with ORDER BY on both columns (see a previous slide)
- Case 2: query without index
  - drop the index
  - execute the same query & record the exec. time
- Observe the execution time improvement.

(D) Project Update

Project update

- Incorporate indexes into your project
  - what columns, what indexes?
- Observe the effect of index
- Update Jira project

Tutorial & Workshop

See Canvas.