

# 实验5：存储与索引

注意：第3步中，需要用到college数据库，请提前创建好并导入数据。

1. 启动MySQL数据库
2. 通过PowerShell连接到MySQL；
3. 执行以下SQL语句，创建一个名为 `mydb` 的数据库，同时在该模式下创建数据表 `mytakes` 和 `mysection`；

```
1 DROP DATABASE IF EXISTS mydb;
2 CREATE DATABASE mydb;
3 USE mydb;
4 DROP TABLE IF EXISTS mytakes;
5 CREATE TABLE `mytakes` (
6   `ID` varchar(5) NOT NULL,
7   `course_id` varchar(8) NOT NULL,
8   `sec_id` varchar(8) NOT NULL,
9   `semester` varchar(6) NOT NULL,
10  `year` int NOT NULL,
11  `grade` varchar(2) DEFAULT NULL,
12  PRIMARY KEY (`ID`,`course_id`,`sec_id`,`semester`,`year`) USING BTREE
13 );
14 insert into mytakes select * from college.takes;
15
16 DROP TABLE IF EXISTS mysection;
17 CREATE TABLE `mysection` (
18   `section_id` int NOT NULL AUTO_INCREMENT,
19   `year` int NOT NULL,
20   `semester` varchar(6) NOT NULL,
21   `building` varchar(15),
22   `room_number` varchar(7),
23   `time_slot_id` varchar(4),
24   `course_id` varchar(8),
25   `sec_id` varchar(8) NOT NULL,
26   PRIMARY KEY (`section_id`)
27 );
28 insert into mysection (`course_id`, `sec_id`, `semester`, `year`, `building`,
   `room_number`, `time_slot_id`) select * from college.section;
```

## 4. INSERT

执行下列语句，观察索引对查询性能的影响，记录每条语句返回的结果并解释其完成的操作；

```
1 USE mydb;
2
3 EXPLAIN insert into mytakes select * from college.takes;
4 EXPLAIN ANALYZE insert into mytakes select * from college.takes;
5
6 EXPLAIN SELECT * FROM mytakes;
7 EXPLAIN SELECT * FROM mytakes LIMIT 1;
8 EXPLAIN ANALYZE SELECT * FROM mytakes;
9 EXPLAIN ANALYZE SELECT * FROM mytakes LIMIT 1;
```

## 5. SELECT with INDEX

执行下列语句，观察索引对查询性能的影响，记录每条语句返回的结果并解释其完成的操作；

mysql> SHOW INDEXES FROM mytakes;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
mytakes	0	PRIMARY	1	ID	A	2000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	2	course_id	A	29254	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	3	sec_id	A	30000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	4	semester	A	30000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	5	year	A	30000	NULL	NULL		BTREE			YES	NULL

5 rows in set (0.02 sec)

```
1 USE mydb;
2
3 SHOW INDEXES FROM mytakes; -- 执行结果要同上图所示，如有多余的index，请先drop删除掉
4 EXPLAIN SELECT * FROM mytakes where course_id='158';
5 EXPLAIN ANALYZE SELECT * FROM mytakes where course_id='158';
6
7 CREATE INDEX course_id ON mytakes(course_id);
8 EXPLAIN SELECT * FROM mytakes where course_id='158';
9 EXPLAIN ANALYZE SELECT * FROM mytakes where course_id='158';
10
11 -- 分析索引失效原因
12 EXPLAIN SELECT * FROM mytakes where course_id=158;
13 EXPLAIN ANALYZE SELECT * FROM mytakes where course_id=158;
14
15 -- 删除索引
16 DROP INDEX course_id ON mytakes;
```

## 6. SELECT with INDEX -- ADVANCED

执行下列语句，观察索引对查询性能的影响，记录每条语句返回的结果并解释其完成的操作；

mysql> SHOW INDEXES FROM mysection;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
mysection	0	PRIMARY	1	section_id	A	100	NULL	NULL		BTREE			YES	NULL

1 row in set (0.01 sec)

## Multiple-Column Indexes/Composite indexes

```
1 USE mydb;
2
3 SHOW INDEXES FROM mysection; -- 执行结果要同上图所示，如有多余的index，请先drop删除掉
4 EXPLAIN SELECT * FROM mysection where year=2006;
5 EXPLAIN ANALYZE SELECT * FROM mysection where year=2006;
6 EXPLAIN SELECT * FROM mysection where year=2006 and semester='Fall';
7 EXPLAIN ANALYZE SELECT * FROM mysection where year=2006 and semester='Fall';
8 EXPLAIN SELECT * FROM mysection where semester='Fall';
9 EXPLAIN ANALYZE SELECT * FROM mysection where semester='Fall';
10
11 CREATE INDEX composite_index on mysection(year,semester,building);
12 EXPLAIN SELECT * FROM mysection where year=2006;
13 EXPLAIN ANALYZE SELECT * FROM mysection where year=2006;
14 EXPLAIN SELECT * FROM mysection where year=2006 and semester='Fall';
15 EXPLAIN ANALYZE SELECT * FROM mysection where year=2006 and semester='Fall';
16 -- 请分析不能使用上面建立的composite_index的原因
17 EXPLAIN SELECT * FROM mysection where semester='Fall';
18 EXPLAIN ANALYZE SELECT * FROM mysection where semester='Fall';
```

## 7. SORT with INDEX

执行下列语句，观察索引对排序性能的影响，记录每条语句返回的结果并解释其完成的操作；

mysql> SHOW INDEXES FROM mytakes;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
mytakes	0	PRIMARY	1	ID	A	2000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	2	course_id	A	29254	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	3	sec_id	A	30000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	4	semester	A	30000	NULL	NULL		BTREE			YES	NULL
mytakes	0	PRIMARY	5	year	A	30000	NULL	NULL		BTREE			YES	NULL

5 rows in set (0.02 sec)

```
1 USE mydb;
2
3 SHOW INDEXES FROM mytakes; -- 执行结果要同上图所示，如有多余的index，请先drop删除掉
4 EXPLAIN SELECT * FROM mytakes ORDER BY course_id DESC LIMIT 10;
5 EXPLAIN ANALYZE SELECT * FROM mytakes ORDER BY course_id DESC LIMIT 10;
6
7 CREATE INDEX course_id ON mytakes(course_id);
8 EXPLAIN SELECT * FROM mytakes ORDER BY course_id DESC LIMIT 10;
9 EXPLAIN ANALYZE SELECT * FROM mytakes ORDER BY course_id DESC LIMIT 10;
```

## 索引性能测试

设计一个用于测试MySQL索引性能的实验，自行定义测试数据集及测试语句(可以编写程序生成测试数据集，或使用下载的公开数据集)

1. 测试维度包括：

- **索引类型**:无索引、唯一索引(btree、hash)、非唯一索引(btree、hash)
- **查询类型**:等值查询、范围查询
- **查询结果集**:单条、少量(<10)、大量(1,000)
- **数据集规模**:100、10,000、1,000,000

2. 对所使用的测试环境、数据集、测试语句和测试方法进行说明;

3. 通过图表方式展示测试结果，并对结果进行分析说明;