Origin

author

id	name	profile
1	kim	developer
2	lee	dba

topic

title	description	created	author_id
MySQL	MySQL is ...	2011	1
ORACLE	ORACLE is ...	2012	1
SQL SERVER	SQL SERVER is	2013	2

topic_tag_relation

topic_title	tag_id
MySQL	1
MySQL	2
ORACLE	1
ORACLE	3

tag

id	name
1	rdb
2	free
3	commercial

topic_type

title	type	price
MySQL	paper	10000
MySQL	online	0
ORACLE	online	15000

실험용 데이터 [1]

컬럼의 역정규화 - 컬럼 중복 : JOIN을 줄이기

목표 **topic_tag_relation.topic_title 와 같이 MySQL 의 태그의 이름을 알고 싶다.**

역정규화 이전 쿼리

```
SELECT
tag.name
FROM topic_tag_relation AS TTR
LEFT JOIN tag
ON TTR.tag_id = tag.id
WHERE topic_title = 'MySQL';
```

역정규화된 테이블 **topic_tag_relation**

topic_title	tag_id	tag_name
MySQL	1	rdb
MySQL	2	free
ORACLE	1	rdb
ORACLE	3	commercial

테이블 변경 쿼리

```
ALTER TABLE `topic_tag_relation` ADD COLUMN `tag_name` VARCHAR(45) NULL AFTER `tag_id`;

UPDATE `topic_tag_relation` SET `tag_name` = 'rdb' WHERE (`topic_title` = 'MySQL') and (`tag_id` = '1');
UPDATE `topic_tag_relation` SET `tag_name` = 'free' WHERE (`topic_title` = 'MySQL') and (`tag_id` = '2');
UPDATE `topic_tag_relation` SET `tag_name` = 'rdb' WHERE (`topic_title` = 'ORACLE') and (`tag_id` = '1');
UPDATE `topic_tag_relation` SET `tag_name` = 'commercial' WHERE (`topic_title` = 'ORACLE') and (`tag_id` = '3');
```

역정규화 이후 쿼리

```
SELECT tag_name FROM topic_tag_relation WHERE topic_title = 'MySQL';
```

컬럼의 역정규화 - 파생 컬럼의 형성 : 계산작업을 줄이기

목표 각각의 저자가 몇개의 글을 작성했는지를 목록으로 표현한다.

역정규화 이전 쿼리

```
SELECT
    author_id, COUNT(author_id)
FROM
    topic
GROUP BY author_id;
```

역정규화 된 표 **author**

id	name	profile	topic_count
1	kim	developer	2
2	lee	dba	1

테이블 변경 쿼리

```
ALTER TABLE `author` ADD COLUMN `topic_count` INT NULL AFTER `profile`;
UPDATE `author` SET `topic_count` = '2' WHERE (`id` = '1');
UPDATE `author` SET `topic_count` = '1' WHERE (`id` = '2');
```

역정규화 후 쿼리

```
SELECT
id, topic_count
FROM
author;
```

테이블의 역정규화 - 컬럼을 기준으로 테이블을 분리

topic

title	created	author_id
MySQL	2011	1
ORACLE	2012	1
SQL SERVER	2013	2

topic_description

title	description
MySQL	MySQL is ...
ORACLE	ORACLE is ...
SQL SERVER	SQL SERVER is ...

테이블의 역정규화 - 행을 기준으로 테이블 분리

topic_1000

title	description	created	author_id
MySQL	MySQL is ...	2011	1
ORACLE	ORACLE is ...	2012	1

topic_1500

title	description	created	author_id
SQL SERVER	SQL SERVER is	2013	1500

관계의 역정규화 - 지름길을 만든다

목표 저자의 태그 아이디와 태그명을 조회한다.

역정규화 이전 쿼리

```
SELECT
    tag.id, tag.name
FROM
    topic_tag_relation AS TTR
LEFT JOIN tag ON TTR.tag_id = tag.id
LEFT JOIN topic ON TTR.topic_title = topic.title
WHERE author_id = 1;
```

역정규화 이후 표 **topic_tag_relation**

topic_title	tag_id	author_id
MySQL	1	1
MySQL	2	1
ORACLE	1	1
ORACLE	3	1

역정규화 쿼리

```
ALTER TABLE `topic_tag_relation` ADD COLUMN `author_id` INT NULL AFTER `tag_name`;
UPDATE `topic_tag_relation` SET `author_id` = '1' WHERE (`topic_title` = 'MySQL') and (`tag_id` = '1');
UPDATE `topic_tag_relation` SET `author_id` = '1' WHERE (`topic_title` = 'MySQL') and (`tag_id` = '2');
UPDATE `topic_tag_relation` SET `author_id` = '1' WHERE (`topic_title` = 'ORACLE') and (`tag_id` = '1');
UPDATE `topic_tag_relation` SET `author_id` = '1' WHERE (`topic_title` = 'ORACLE') and (`tag_id` = '3');
```

역정규화 이후 쿼리

```
SELECT
    tag.id, tag.name
FROM
    topic_tag_relation AS TTR
LEFT JOIN tag ON TTR.tag_id = tag.id
WHERE TTR.author_id = 1;
```

```
[1] DROP TABLE IF EXISTS `author`;
CREATE TABLE `author` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `profile` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `author` VALUES (1,'kim','developer'),(2,'lee','DBA');

DROP TABLE IF EXISTS `tag`;
CREATE TABLE `tag` (
  `id` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `tag` VALUES (1,'rdb'),(2,'free'),(3,'commercial');

DROP TABLE IF EXISTS `topic`;
CREATE TABLE `topic` (
  `title` varchar(50) NOT NULL,
  `description` text,
  `created` datetime DEFAULT NULL,
  `author_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`title`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `topic` VALUES ('MySQL','MySQL is ...','2011-01-01 00:00:00',1),('ORACLE','ORACLE is
...','2012-02-03 00:00:00',1),('SQL Server','SQL Server is ..','2013-01-04 00:00:00',2);
DROP TABLE IF EXISTS `topic_tag_relation`;

CREATE TABLE `topic_tag_relation` (
  `topic_title` varchar(50) NOT NULL,
  `tag_id` int(11) NOT NULL,
  PRIMARY KEY (`topic_title`,`tag_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `topic_tag_relation` VALUES ('MySQL',1),('MySQL',2),('ORACLE',1),('ORACLE',3);

DROP TABLE IF EXISTS `topic_type`;
CREATE TABLE `topic_type` (
  `title` varchar(45) NOT NULL,
  `type` char(6) NOT NULL,
  `price` int(11) DEFAULT NULL,
  PRIMARY KEY (`title`,`type`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `topic_type` VALUES ('MySQL','online',0),('MySQL','paper',10000),('ORACLE','online',
15000);
```