-- 1/ Select movie and roles count

SELECT title,

(SELECT COUNT(\*) FROM role WHERE role.movie\_id=movie.id GROUP BY role.movie\_id) as rcount

FROM movie;

ALTER TABLE role ADD KEY (movie\_id);

-- 2/ The same but with join

SELECT title, COUNT(\*) as rcount FROM movie LEFT JOIN role ON movie.id=role.movie\_id GROUP BY movie.id;

-- 3/ Movie with highest fb likes

select \* from movie where id = (select movie\_id from review order by facebook\_likes desc limit 1);

-- 4/ Find duplcates

select \* from movie m1 where (select count(\*) from movie m2 where m2.title = m1.title) >= 2;

-- 5/ Find moview without error

select \* from movie m where (select count(\*) from role r where r.movie\_id=m.id)=0;

insert into movie set title="xxx";

select \* from movie m where (select count(\*) from role r where r.movie\_id=m.id)=0;

alter table role add key (movie\_id);

-- 6/ nother way

select \* from movie m where not exists (select \* from review r where r.movie\_id=m.id);

-- 7/ average roles amount

select avr(sumcol) from (SELECT count(\*) as sumcol from role group by movie\_id) as role\_count;

-- 1/ Inner

select m.\*, r.\* from movie m INNER JOIN review r ON m.id = r.movie\_id ORDER BY m.id;

-- 2/ Left

select m.\*, r.\* from movie m LEFT JOIN review r ON m.id = r.movie\_id ORDER BY m.id;

select m.\*, r.\* from movie m LEFT JOIN review r ON m.id = r.movie\_id WHERE r.id IS NULL ORDER BY m.id;

-- 3/ Full Outer

SELECT \* FROM movie

LEFT JOIN review ON movie.id = review.movie\_id

UNION

SELECT \* FROM movie

RIGHT JOIN review ON movie.id = review.movie\_id;

-- 4/ Multiple

CREATE TABLE `products` (

`id` int(11),

`title` varchar(255),

`created\_at` datetime

);

CREATE TABLE `product\_options` (

`id` int(11),

`name` varchar(255)

);

CREATE TABLE `product2options` (

`product\_id` int(11),

`option\_id` int(11),

`value` int(11)

);

INSERT INTO `products` (`id`, `title`, `created\_at`) VALUES

(1, 'Cup', '2016-12-17 20:00:00'),

(2, 'Spoon', '2017-01-18 20:00:00'),

(3, 'Plate', '2017-01-19 20:00:00');

INSERT INTO `product\_options` (`id`, `name`) VALUES

(11, 'Weight'),

(12, 'Capacity');

INSERT INTO `product2options` (`product\_id`, `option\_id`, `value`) VALUES

(1, 11, 200),

(1, 12, 250),

(2, 11, 35),

(2, 12, 15),

(3, 11, 310),

(3, 12, 300),

(2, 11, 45),

(2, 12, 25);

select distinct p.\*, po.\*, p2o.\* from products p INNER JOIN product2options p2o ON p.id=p2o.product\_id JOIN product\_options po ON p2o.option\_id=po.id where created\_at > '2017-01-01' and name='Weight' and value='310';

-- 1/ Add date field and fill it in with data

ALTER TABLE review ADD column created\_at DATETIME NULL;

UPDATE review SET created\_at = FROM\_UNIXTIME(UNIX\_TIMESTAMP('2015-01-01 00:00:00') + FLOOR(0 + (RAND() \* 63072000)));

-- 2/ SELECT, we cannot use index for rdate

SELECT DATE(created\_at) AS rdate, COUNT(\*) AS rcount FROM review GROUP BY rdate;

-- 3/ ADD virtuall column and index

ALTER TABLE review ADD COLUMN created\_at\_date DATE GENERATED ALWAYS AS (DATE(created\_at)) VIRTUAL;

ALTER TABLE review ADD KEY (created\_at\_date);

-- 4/ Now index is used

SELECT created\_at\_date, COUNT(\*) AS rcount FROM review GROUP BY created\_at\_date;

-- 5/ Which one

ALTER TABLE review ADD KEY date\_movie (created\_at\_date, movie\_id);

ALTER TABLE review ADD KEY movie\_date (movie\_id, created\_at\_date);