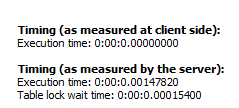
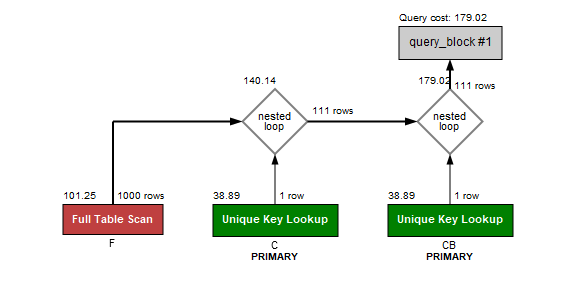
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №5  Индексы, хранимые процедуры, представления | Ф.И.О. | Куканов А.Д. |
| Группа | ИВТ-263 |
| Преподаватель | Соколов А.А. |
| Дата сдачи |  |

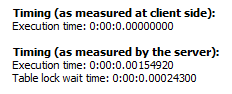
1. SELECT C.`name` as `Customer`, CB.`name` as `Comic book`, F.`text` as `review`, F.`mark` FROM `customer` C   
INNER JOIN `feedback` F ON C.`id` = F.`customer\_id`   
INNER JOIN `comic\_book` CB ON F.`comic\_book\_id` = CB.`id`  
WHERE F.`date` between '2021-06-06' AND '2021-07-07';

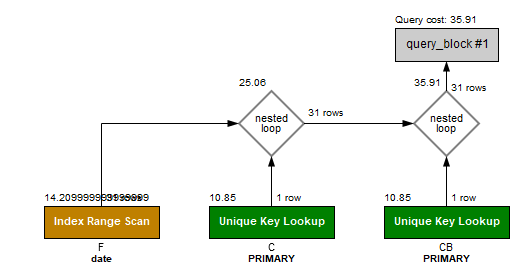
Без индексов:





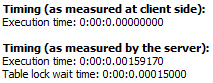
С индексом:

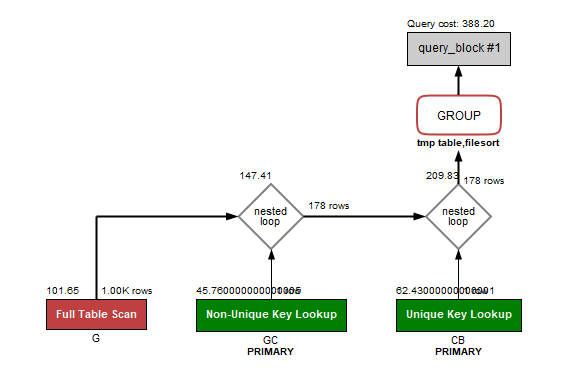




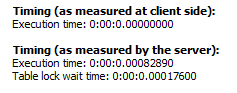
2. SELECT CB.`name`, group\_concat(G.`name`) as `genre\_name` FROM `comic\_book` CB INNER JOIN `genres\_of\_comic` GC ON GC.`comic\_book\_id` = CB.`id` INNER JOIN `genre` G ON G.`id` = GC.`genre\_id` WHERE G.`name` LIKE ('thriller%') GROUP BY CB.`name` ;

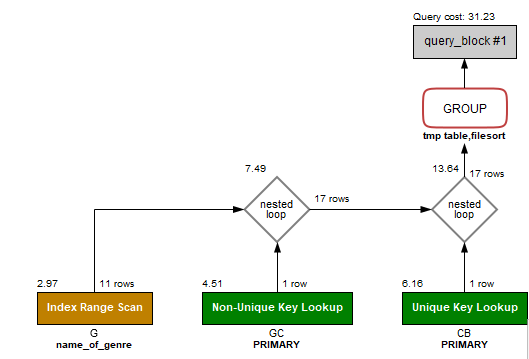
Без индексов:





С индексом:





3. SELECT CB.`name`, CB.`Format`, CB.`description`, CB.`Publishing\_date`, CB.`FOC`, AVG (Mark) as `average mark`,GROUP\_CONCAT( CA.`name`) as `Artist`, GROUP\_CONCAT(CVA.`name`) as `cover artist` , GROUP\_CONCAT(CW.`name`) as `Writer`, L.`name` as `label`, PB.`name` as `publisher` FROM

`feedback` FB JOIN `comic\_book` CB ON FB.`comic\_book\_id` = CB.`id`

JOIN `artists\_of\_comic` AOC ON AOC.`comic\_book\_id` = CB.`id`

JOIN `comic\_artist` CA ON CA.`id` = AOC.`comic\_artist\_id`

JOIN `writers\_of\_comic\_book` WOC ON WOC.`comic\_book\_id` = CB.`id`

JOIN `comic\_writer` CW ON CW.`id` = WOC.`comic\_writer\_id`

JOIN `comic\_cover` CC ON CC.`comic\_book\_id` = CB.`id`

JOIN `cover\_artist` CVA ON CVA.`id` = CC.`cover\_artist\_id`

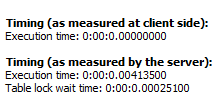
JOIN `label` L ON L.`id` = CB.`label\_id`

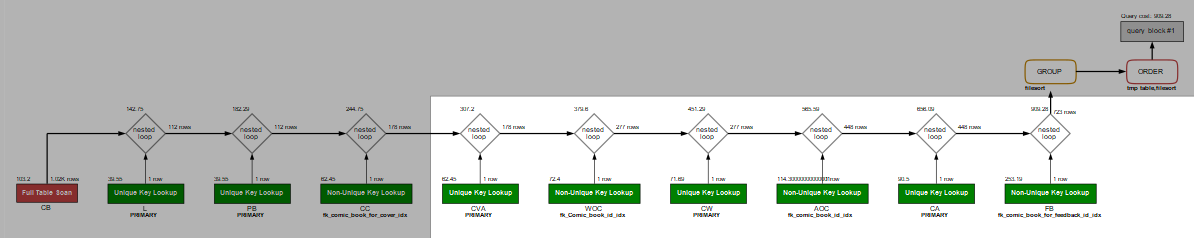
JOIN `publisher` PB ON PB.`id` = L.`publisher\_id`

WHERE CB.`publishing\_date` BETWEEN '2021-01-06' AND '2021-11-07'

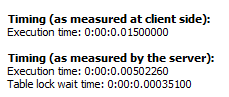
GROUP BY CB.`name` ORDER BY AVG (Mark) DESC LIMIT 5;

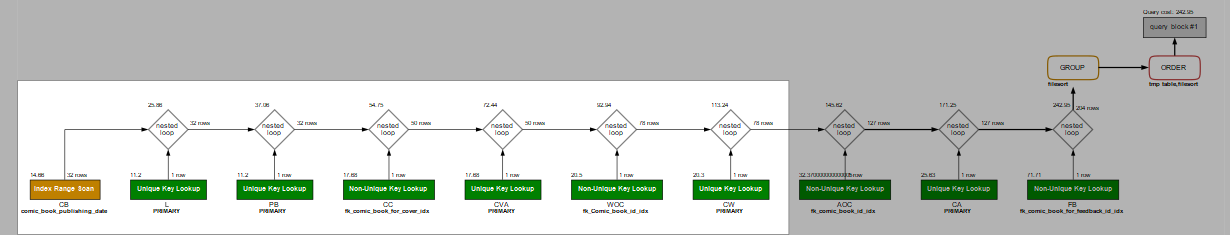
Без индексов:





С индексом:





4. SELECT CS.`adress`, CB.`name`, CB.`Format`, CB.`description`, CB.`Publishing\_date`, CB.`FOC`, L.`name` as `Label`, AV.`count`, AV.`price`, AV.`discount` FROM `comic\_shop` CS

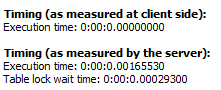
JOIN `shipment` SH ON SH.`comic\_shop\_id` = CS.`id`

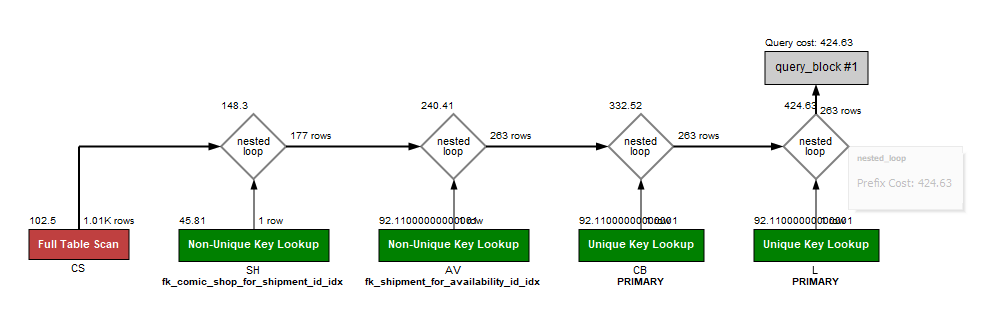
JOIN `availability` AV ON AV.`shipment\_id` = SH.`id`

JOIN `comic\_book` CB ON AV.`comic\_book\_id` = CB.`id`

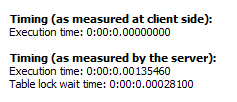
JOIN `label` L ON CB.`label\_id` = L.`id` WHERE CS.`adress` LIKE '709%';

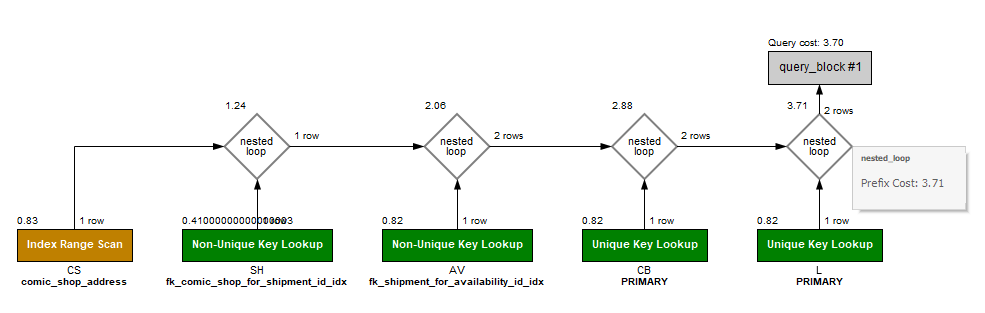
Без индекса:



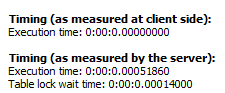


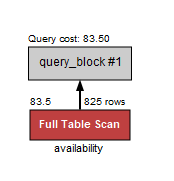
С индексом:



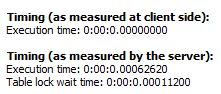


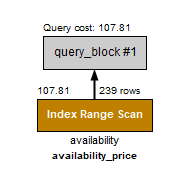
5. SELECT `price`,`shipment\_id` FROM `availability` WHERE `price` > 3500;





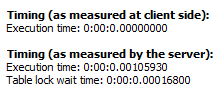
С индексом:

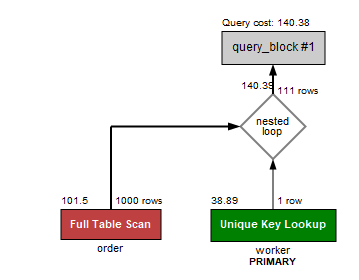




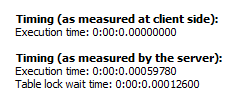
6. SELECT `order`.`date`, `worker`.`name` FROM `worker` JOIN `order` ON `worker`.`id` = `order`.`worker\_id` WHERE `order`.`date` BETWEEN '2021-12-01' AND '2022-02-02';

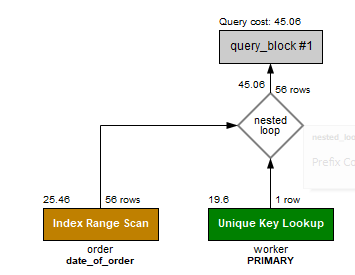
Без индексов:





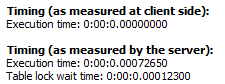
С индексом:

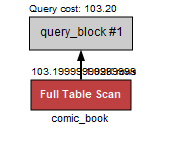




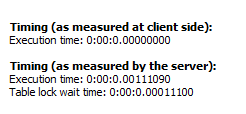
7. SELECT `id` FROM `comic\_book` WHERE `FOC`> '2021-08-08' ;

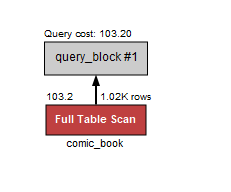
Без индексов:





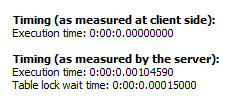
С индексами:

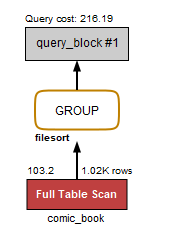




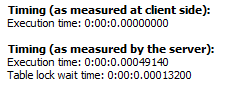
8. SELECT `Publishing\_date`, Group\_concat(`name`) as `Comic book` FROM `comic\_book` GROUP BY `publishing\_date`;

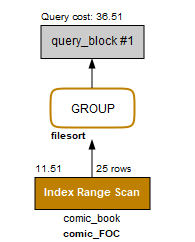
Без индекса:





С индексом:





Добавление индексов:

1 CREATE INDEX `feedback\_date` ON feedback(date);

2 CREATE INDEX `genre\_name` ON genre(name);

3 CREATE INDEX `comic\_book\_publishing\_date` ON comic\_book(publishing\_date);

4 CREATE INDEX `comic\_shop\_address` ON comic\_shop(adress);

5 CREATE INDEX `availability\_price` ON `availability`(price);

6 CREATE INDEX `date\_of\_order` ON `order`(date);

7 CREATE INDEX `comic\_book\_FOC` ON `comic\_book`(`FOC`);

8 CREATE INDEX `comic\_FOC` ON `comic\_book`(`FOC`);

**Процедуры**:

1. Нечеткая оценка комикса по отзывам:

DELIMITER $$

CREATE PROCEDURE `Comics\_mark`()

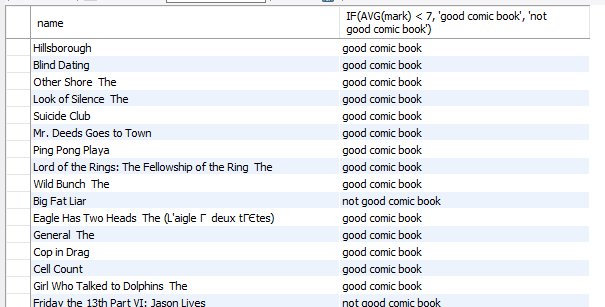
BEGIN

SELECT CB.`name`, IF(AVG(mark) < 7, 'good comic book', 'not good comic book') FROM `feedback` F

INNER JOIN `comic\_book` CB ON F.`comic\_book\_id` = CB.`id` GROUP BY CB.`name`;

END$$

CALL Comics\_mark();



1. Нечеткая оценка стоимости заказов

PROCEDURE `Shipment\_cost`()

BEGIN

SELECT O.`id` as `order\_id`,

CASE

WHEN price\*AV.count = 3500 THEN 'Free shipment'

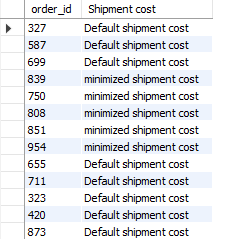
WHEN price\*AV.count < 3500 THEN 'Default shipment cost'

WHEN price\*AV.count > 3500 THEN 'minimized shipment cost'

END AS `Shipment cost`

FROM `order` O INNER JOIN `order\_availability` OA ON O.`id` = OA.`order\_id` INNER JOIN `availability` AV ON OA.`availability\_id` = AV.`id` GROUP BY O.`id`;

END



1. Нечёткая оценка поставок в магазин

PROCEDURE `Shipment\_mark`()

BEGIN

SELECT CS.`adress`,

CASE

WHEN SUM(count) > 500 THEN 'Крупная поставка'

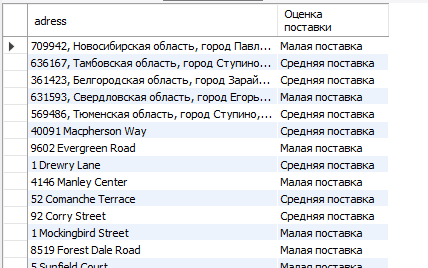
WHEN SUM(count) < 100 THEN 'Малая поставка'

ELSE 'Средняя поставка'

END AS `Оценка поставки`

FROM comic\_shop CS INNER JOIN SHIPMENT SH ON CS.`id` = SH.`comic\_shop\_id` INNER JOIN availability AV ON SH.`id` = AV.`shipment\_id` GROUP BY CS.`adress`;

END



**Функции**:

1. Стоимость комиксов в наличии

FUNCTION `Shipment\_cost`(count int, price int) RETURNS int

DETERMINISTIC

BEGIN

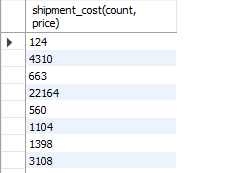
DECLARE scost INT;

SET scost = 0;

SELECT count\*price into scost FROM availability LIMIT 1;

RETURN scost;

END



2)

Нечеткая оценка стоимости методов доставки

FUNCTION `delivery\_high\_price`(`price` int) RETURNS int

DETERMINISTIC

BEGIN

DECLARE isHigh INTEGER;

SET `count` = 0;

IF `price` > 200 THEN

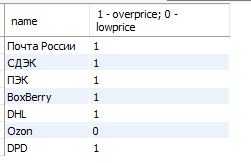
SET `isHigh` = ` isHigh ` + 1;

END IF;

RETURN `count`;

END

select name, delivery\_high\_price(price) as `1 - overprice; 0 - lowprice` from delivery\_method;



3) Подсчёт крупных поставок

FUNCTION `huge\_shipments`(`price` int, `count` int) RETURNS int

DETERMINISTIC

BEGIN

DECLARE sum\_price INTEGER;

if `count` > 15 THEN

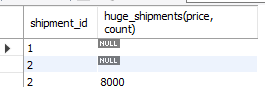
set sum\_price = `price` \* `count`;

END IF;

return sum\_price;

END

select shipment\_id,huge\_shipments(price, count) from availability;



**Представления**

1. Персонал магазина в аризоне

VIEW `comic\_shop\_web`.`arizona\_shop\_staff` AS

SELECT

`w`.`Name` AS `name`, `w`.`Post` AS `post`

FROM

(`comic\_shop\_web`.`worker` `w`

JOIN `comic\_shop\_web`.`comic\_shop` `cs` ON ((`w`.`Comic\_shop\_ID` = `cs`.`ID`)))

WHERE

(`cs`.`Adress` LIKE '%Arizona%')

select \* from arizona\_shop\_staff;

1. Перечень пунктов выдачи

VIEW `comic\_shop\_web`.`points` AS

SELECT

`dm`.`Name` AS `name`, `pup`.`Adress` AS `adress`

FROM

(`comic\_shop\_web`.`pick\_up\_point` `pup`

JOIN `comic\_shop\_web`.`delivery\_method` `dm` ON ((`pup`.`Delivery\_method\_ID` = `dm`.`ID`)))

select\*from points;

1. Комиксы и их издатели

VIEW `comic\_shop\_web`.`comic\_books` AS

SELECT

`cb`.`Name` AS `name`,

`cb`.`Format` AS `format`,

`cb`.`Publishing\_date` AS `publishing\_date`,

`cb`.`FOC` AS `FOC`,

`l`.`Name` AS `label`,

`p`.`Name` AS `publisher`

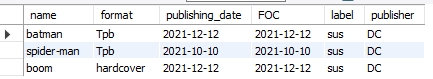
FROM

((`comic\_shop\_web`.`comic\_book` `cb`

JOIN `comic\_shop\_web`.`label` `l` ON ((`cb`.`Label\_ID` = `l`.`ID`)))

JOIN `comic\_shop\_web`.`publisher` `p` ON ((`l`.`Publisher\_ID` = `p`.`ID`)))

select\*from comic\_books;



Модификация:

Процедура на вход id комикса которая возвращает через out параметры список писателей и список художников и художники

CREATE DEFINER=`root`@`localhost` PROCEDURE `mod`(in comic\_id int, out cb\_name varchar(255), out writer varchar(255), out artist varchar(255), out cover\_artist varchar(255))

BEGIN

select cb.`name`, group\_concat(distinct CW.`name`), group\_concat(distinct CA.`name`), group\_concat(distinct CvA.`name`)

INTO cb\_name, writer, artist, cover\_artist

from `comic\_book` CB

inner join `artists\_of\_comic` AOC ON CB.`ID` = AOC.`comic\_book\_ID`

inner join `comic\_artist` CA on AOC.`comic\_artist\_id` = CA.id

INNER JOIN `writers\_of\_comic\_book` WOC ON WOC.`comic\_book\_id` = CB.id

INNER JOIN `comic\_writer` CW ON WOC.`comic\_writer\_id` = CW.id

INNER JOIN `comic\_cover` CC ON CC.`comic\_book\_id` = CB.id

INNER JOIN `cover\_artist` CvA ON CvA.`id` = CC.`cover\_artist\_id` where CB.id = comic\_id GROUP BY CB.name;

END

CALL `mod`(1, @name, @writer, @artist,@cover\_artist);

select @name, @writer, @artist,@cover\_artist;

