## Запрос №1

### Реализация

SELECT Н\_ВЕДОМОСТИ.ИД, Н\_ОЦЕНКИ.КОД

FROM Н\_ОЦЕНКИ

RIGHT JOIN Н\_ВЕДОМОСТИ ON Н\_ОЦЕНКИ.КОД=Н\_ВЕДОМОСТИ.ОЦЕНКА

WHERE Н\_ОЦЕНКИ.КОД='неявка'

AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД<153285;

### Уменьшение времени выполнения запроса

#### Индексы

Для таблицы Н\_ВЕДОМОСТИ:

* Индекс B-tree для ЧЛВК\_ИД для сравнения строк (Н\_ВЕДОМОСТИ.ЧЛВК\_ИД<153285)
* Индекс Hash для ОЦЕНКА для объединения таблиц (ON Н\_ОЦЕНКИ.КОД=Н\_ВЕДОМОСТИ.ОЦЕНКА)

Для таблицы Н\_ОЦЕНКИ:

* Индекс Hash для КОД для проверки на равенство у строк (Н\_ОЦЕНКИ.КОД='неявка') и объединения таблиц (ON Н\_ОЦЕНКИ.КОД=Н\_ВЕДОМОСТИ.ОЦЕНКА)

### Возможные планы выполнения запросов (без индексов)

#### План №1

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

#### План №2

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

#### Оптимальный план

План №2 является оптимальным, потому что фильтрация выполняется на меньшем объеме данных, так как она вызвана до объединения таблиц

### Изменения планов при добавлении индексов

Будет выполняться индексный скан таблицы, вместо полного

Nested Loop Join выполнится быстрее, из-за индексах на атрибутах Н\_ОЦЕНКИ.КОД и Н\_ВЕДОМОСТИ.ОЦЕНКА

### План выполнения 1 запроса

QUERY PLAN

---------------------------------------------------------------------------------------------------------------------

Nested Loop (cost=7.69..1252.34 rows=405 width=38) (actual time=0.167..0.641 rows=392 loops=1)

-> Seq Scan on "Н\_ОЦЕНКИ" (cost=0.00..1.11 rows=1 width=34) (actual time=0.016..0.017 rows=1 loops=1)

Filter: (("КОД")::text = 'неявка'::text)

Rows Removed by Filter: 8

-> Bitmap Heap Scan on "Н\_ВЕДОМОСТИ" (cost=7.69..1247.18 rows=405 width=10) (actual time=0.147..0.572 rows=392 loops=1)

Recheck Cond: (("ОЦЕНКА")::text = 'неявка'::text)

Filter: ("ЧЛВК\_ИД" < 153285)

Heap Blocks: exact=150

-> Bitmap Index Scan on "ВЕД\_ОЦЕНКА\_I" (cost=0.00..7.59 rows=423 width=0) (actual time=0.116..0.116 rows=392 loops=1)

Index Cond: (("ОЦЕНКА")::text = 'неявка'::text)

Planning Time: 0.785 ms

Execution Time: 0.725 ms

## Запрос №2

### Реализация

SELECT Н\_ЛЮДИ.ИМЯ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, Н\_СЕССИЯ.ЧЛВК\_ИД

FROM Н\_ЛЮДИ

LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

LEFT JOIN Н\_СЕССИЯ ON Н\_СЕССИЯ.ЧЛВК\_ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИМЯ='Ярослав'

AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД='117219'

AND Н\_СЕССИЯ.УЧГОД>'2003/2004';

### Уменьшение времени выполнения запроса

#### Индексы

Для таблицы Н\_ЛЮДИ:

* Индекс Hash для ИМЯ для проверки на равенство у строк (Н\_ЛЮДИ.ИМЯ='Ярослав')
* Индекс Hash для ИД для объединения таблиц (ON Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД)

Для таблицы Н\_ВЕДОМОСТИ:

* Индекс Hash для ЧЛВК\_ИД для проверки на равенство у строк (Н\_ВЕДОМОСТИ.ЧЛВК\_ИД='117219') и объединения таблиц (ON Н\_СЕССИЯ.ЧЛВК\_ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД, ON Н\_ЛЮДИ.ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД)

Для таблицы Н\_СЕССИЯ:

* Индекс B-tree для УЧГОД для сравнения строк (Н\_СЕССИЯ.УЧГОД>'2003/2004')
* Индекс Hash для ЧЛВК\_ИД для объединения таблиц (ON Н\_СЕССИЯ.ЧЛВК\_ИД=Н\_ВЕДОМОСТИ.ЧЛВК\_ИД)

### Возможные планы выполнения запросов (без индексов)

#### План №1

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

#### План №2

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

#### Оптимальный план

План №2 является оптимальным, потому что фильтрация выполняется на меньшем объеме данных, так как она вызвана до объединения таблиц

### Изменения планов при добавлении индексов

Будет выполняться индексный скан таблицы, вместо полного

Nested Loop Join выполнится быстрее, из-за индексах на атрибутах Н\_СЕССИЯ.ЧЛВК\_ИД, Н\_СЕССИЯ.ЧЛВК\_ИД и Н\_ЛЮДИ.ИД

### План выполнения 2 запроса

QUERY PLAN

---------------------------------------------------------------------------------------------------------------------

Nested Loop (cost=4.87..30.70 rows=130 width=21) (actual time=0.100..0.101 rows=0 loops=1)

-> Index Only Scan using "ВЕД\_ЧЛВК\_FK\_IFK" on "Н\_ВЕДОМОСТИ" (cost=0.29..9.43 rows=65 width=4) (actual time=0.042..0.046 rows=31 loops=1)

Index Cond: ("ЧЛВК\_ИД" = 117219)

Heap Fetches: 0

-> Materialize (cost=4.58..19.65 rows=2 width=21) (actual time=0.002..0.002 rows=0 loops=31)

-> Nested Loop (cost=4.58..19.64 rows=2 width=21) (actual time=0.039..0.040 rows=0 loops=1)

-> Index Scan using "ЧЛВК\_PK" on "Н\_ЛЮДИ" (cost=0.28..8.30 rows=1 width=17) (actual time=0.039..0.039 rows=0 loops=1)

Index Cond: ("ИД" = 117219)

Filter: (("ИМЯ")::text = 'Ярослав'::text)

Rows Removed by Filter: 1

-> Bitmap Heap Scan on "Н\_СЕССИЯ" (cost=4.30..11.32 rows=2 width=4) (never executed)

Recheck Cond: ("ЧЛВК\_ИД" = 117219)

Filter: (("УЧГОД")::text > '2003/2004'::text)

-> Bitmap Index Scan on "SYS\_C003500\_IFK" (cost=0.00..4.29 rows=2 width=0) (never executed)

Index Cond: ("ЧЛВК\_ИД" = 117219)

Planning Time: 0.653 ms

Execution Time: 0.151 ms