# Лабораторные работы. Индексы

Для выполнения 1-го задания создадим таблицу с большим количеством записей.

create table TEST\_IDX as select owner, object\_Name,object\_id,object\_type from all\_objects;

Повторим вставку несколько раз для разных значений object\_id+N, где N= 10000,20000,… 60000.

insert into TEST\_IDX select owner, object\_Name,object\_id+60000,object\_type from all\_objects;

commit;

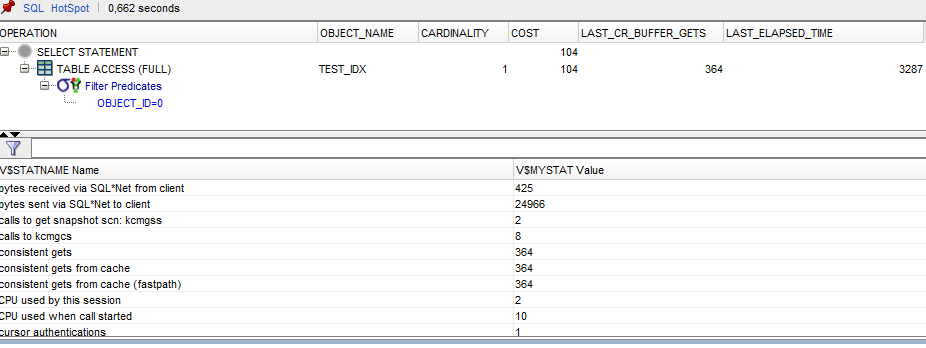
Соберем статистику.

exec dbms\_stats.gather\_schema\_stats('TEST\_1'); TEST\_1 это схема, имя пользователя.

## Назначение индекса

Выполним запрос и простотрим трассировку.

select \* from test\_idx where object\_id=0;

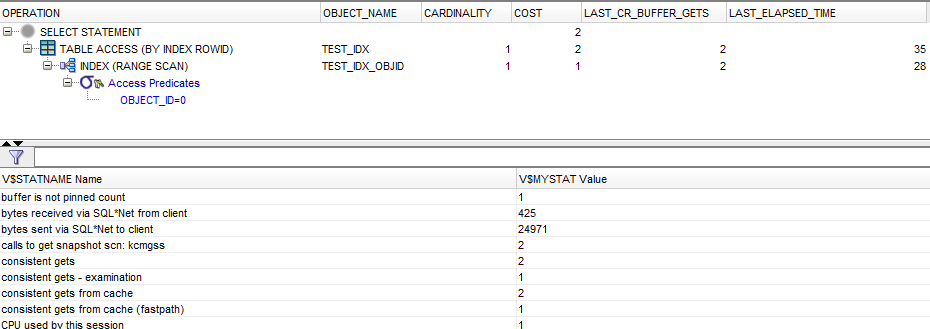


Для выборки одной строки была просканирована вся таблица.

Создадим индекс без регистрации статистики

Create index test\_idx\_objId on test\_idx(object\_id) **nologging compute statistics**;

Повторим предыдущий запрос и просмотрим трассировку



1. **Какие отличия в результатах трассировки?**

***Для выполнения дальнейших заданий, необходимо создать тестовые таблицы из схемы SH (sales\_history). Файлы с кодом и данными прилагаются.***

***Указание по подготовке таблиц:***

* ***создать и заполнить таблицу SHCountries;***
* ***создать и заполнить таблицу SHCustomers.***

## В-деревья.

Выполним запрос.

SELECT /\*+ FULL(c) \*/ c.\*

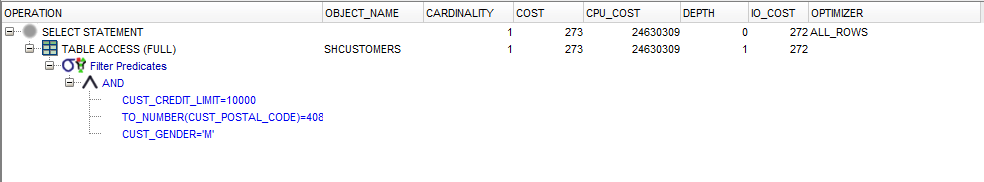
FROM SHcustomers c

WHERE cust\_gender = 'M'

AND cust\_postal\_code = 40804

AND cust\_credit\_limit = 10000

И рассмотрим трассировку:



1. **Что получили?**

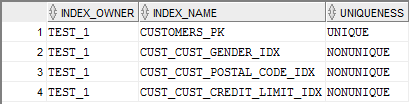
Создадим индексы:

CREATE INDEX cust\_cust\_gender\_idx ON SHcustomers(cust\_gender) NOLOGGING COMPUTE STATISTICS;

CREATE INDEX cust\_cust\_postal\_code\_idx ON SHcustomers(cust\_postal\_code) NOLOGGING COMPUTE STATISTICS;

CREATE INDEX cust\_cust\_credit\_limit\_idx ON SHcustomers(cust\_credit\_limit) NOLOGGING COMPUTE STATISTICS;

И получим информацию об индексах таблицы ShCustomers



Запустим мониторинг за использование индексов.

ALTER INDEX CUSTOMERS\_PK MONITORING USAGE;

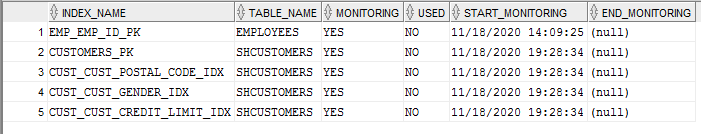
ALTER INDEX CUST\_CUST\_POSTAL\_CODE\_IDX MONITORING USAGE;

ALTER INDEX CUST\_CUST\_GENDER\_IDX MONITORING USAGE;

ALTER INDEX CUST\_CUST\_CREDIT\_LIMIT\_IDX MONITORING USAGE;

И просмотрим исходное состояние

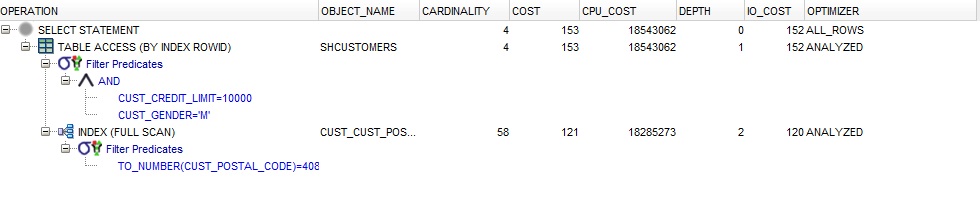
select \* from v$object\_usage;



SELECT /\*+ INDEX(c) \*/ c.\* FROM SHcustomers c

WHERE cust\_gender = 'M' AND cust\_postal\_code = 40804

AND cust\_credit\_limit = 10000



1. **Что получили?**

Использование битовых индексов НЕ РАБОТАЕТ для 11.0.2

SELECT /\*+ INDEX\_COMBINE(c) \*/ c.\*

FROM Shcustomers c

WHERE cust\_gender = 'M'

AND cust\_postal\_code = 40804

AND cust\_credit\_limit = 10000;

select \* from v$parameter order by name;

### Многоключевой индекс

Удалим индексы

* Сформируем список для удаления:

SELECT 'drop index '||i.index\_name||';'

FROM user\_indexes i

WHERE i.table\_name = 'SHCUSTOMERS'

AND NOT EXISTS

(SELECT 'x'

FROM user\_constraints c

WHERE c.index\_name = i.index\_name

AND c.table\_name = i.table\_name

AND c.status = 'ENABLED');

* И удалим.

drop index CUST\_CUST\_GENDER\_IDX;

drop index CUST\_CUST\_POSTAL\_CODE\_IDX;

drop index CUST\_CUST\_CREDIT\_LIMIT\_IDX;

Создадим индекс по трем полям:

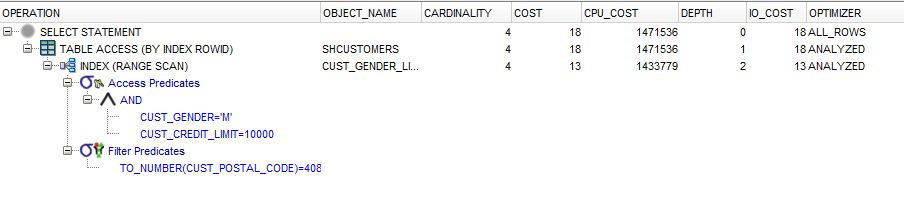
CREATE INDEX cust\_gender\_limit\_code\_idx ON SHcustomers(cust\_gender,cust\_credit\_limit,cust\_postal\_code) NOLOGGING COMPUTE STATISTICS;

Выполним запрос и просмотрим план исполнения. Сравним результатты для разных вариантов условия выборки данных.

1. SELECT /\*+ INDEX(c) \*/ c.\* FROM SHcustomers c WHERE cust\_gender = 'M'

AND cust\_postal\_code = 40804

AND cust\_credit\_limit = 10000;



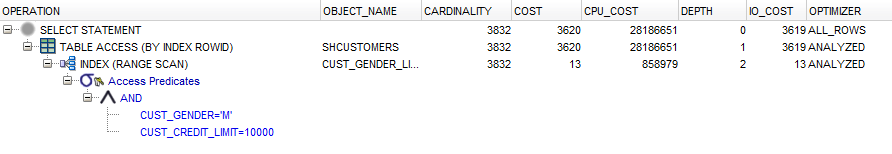
Изменим условие выбора и просмотрим план выполнения запроса:

1. SELECT /\*+ INDEX(c cust\_gender\_limit\_code\_idx) \*/ c.\*

FROM Shcustomers c

WHERE cust\_gender = 'M'

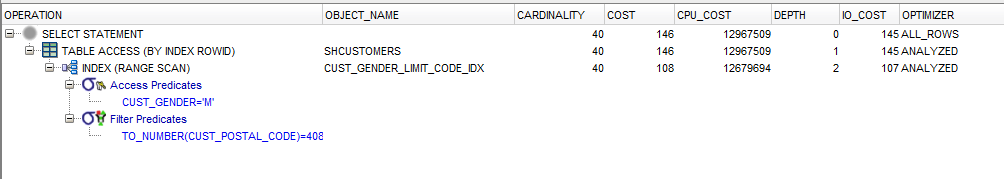
AND cust\_credit\_limit = 10000;



1. **Что получили? (сравнить COST, CARDINALITY)**

Еще вариант изменения в запросе.

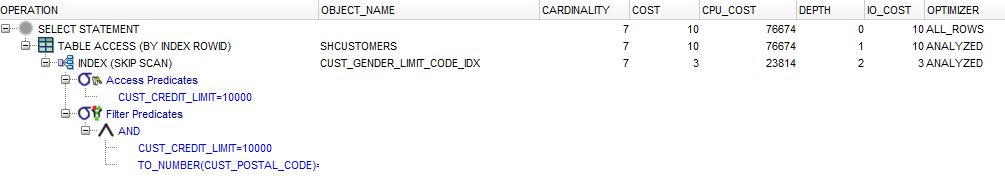
1. SELECT /\*+ INDEX(c cust\_gender\_limit\_code\_idx) \*/ c.\* FROM SHcustomers c WHERE cust\_gender = 'M' AND cust\_postal\_code = 40804;



1. **Что получили? (сравнить COST, CARDINALITY)**

И последний вариант.

1. SELECT /\*+ INDEX(c) \*/ c.\* FROM SHcustomers c WHERE cust\_postal\_code = 40804 AND cust\_credit\_limit = 10000;



1. **Что получили? (сравнить COST, CARDINALITY)**

**Указать причины расхождений с статистике выполнения.**

## BitMap ‑индексы

Удалим все индексы, созданные ранее, и создадим индексы организованные как BitMap по полям фразы WHERE.

CREATE BITMAP INDEX cust\_cust\_gender\_bidx ON SHcustomers(cust\_gender) NOLOGGING COMPUTE STATISTICS;

CREATE BITMAP INDEX cust\_cust\_postal\_code\_bidx ON SHcustomers(cust\_postal\_code) NOLOGGING COMPUTE STATISTICS;

CREATE BITMAP INDEX cust\_cust\_credit\_limit\_bidx ON SHcustomers(cust\_credit\_limit)NOLOGGING COMPUTE STATISTICS;

**В Версии Oracle 11.2.0 не работает!!**

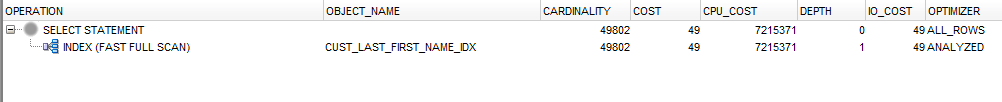
## Использование индексов для выборки данных

Создадим индекс по ФИО клиента.

CREATE INDEX cust\_last\_first\_name\_idx ON SHcustomers(cust\_last\_name,cust\_first\_name) NOLOGGING COMPUTE STATISTICS;

Выполним запрос в котором выполним вывод значения полей в следующем порядке: **cust\_last\_name, cust\_first\_name**.

SELECT c.cust\_last\_name, c.cust\_first\_name FROM SHcustomers c;



1. **Что получили? (что будет, если прядок следования полей изменить или добавить еще поля)**

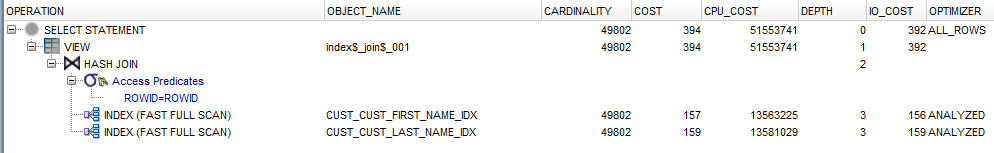
Удалим созданный ранее индекс cust\_last\_first\_name\_idx и создадим два новых по отдельности для каждого поля **cust\_last\_name, cust\_first\_name**.

CREATE INDEX cust\_cust\_last\_name\_idx ON SHcustomers(cust\_last\_name) NOLOGGING COMPUTE STATISTICS;

CREATE INDEX cust\_cust\_first\_name\_idx ON SHcustomers(cust\_first\_name) NOLOGGING COMPUTE STATISTICS;

Сформируем план выполнения запроса:

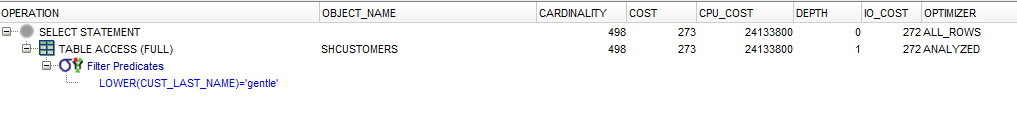
SELECT /\*+ INDEX\_JOIN(c cust\_cust\_first\_name\_idx cust\_cust\_last\_name\_idx) \*/ c.cust\_last\_name, c.cust\_first\_name FROM SHcustomers c;



1. **Что получили? (сравнить с планом выполнения предыдущего запроса)**

Удалим все ранее созданные индексы, кроме CUSTOMERS\_PK и cust\_cust\_last\_name\_idx и сформируем план выполнения запроса:

SELECT cust\_id, country\_id FROM SHcustomers WHERE LOWER( cust\_last\_name) LIKE 'gentle';

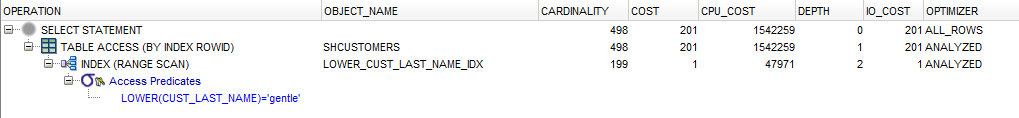


1. **Что получили? (почему индекс не используется?)**

Создадим индекс, основанный на функции LOWER().

CREATE INDEX lower\_cust\_last\_name\_idx ON SHcustomers(LOWER(cust\_last\_name));

И рассмотрим новый план выполнения предыдущего запроса.



1. **Что получили? (сравнить оба плана)**

## Сравнение индексов и таблиц, организованных по индексу.

Создадим таблицу из схемы SH SH promotions. Файлы для создания и заполнения прилагаются.

CREATE table promotions\_iot (promo\_id number primary key, promo\_name VARCHAR2(40), promo\_subcategory VARCHAR2 (30), promo\_category VARCHAR2 (30) , promo\_cost NUMBER, promo\_begin\_date DATE, promo\_end\_date DATE) ORGANIZATION INDEX;

И заполним ее.

INSERT INTO promotions\_iot

SELECT promo\_id, promo\_name, promo\_subcategory, promo\_category, promo\_cost, promo\_begin\_date, promo\_end\_date

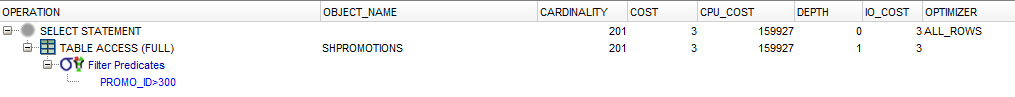
FROM Shpromotions;

Выполним несколько запросов к таблице Shpromotions.

1. SELECT /\*+ NO\_INDEX(s) \*/ \* FROM SHpromotions s WHERE promo\_id > 300;

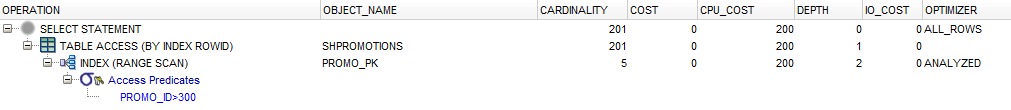
Для примера важно не использовать индекс PROMO\_PK, поэтому отключим его (/\*+ NO\_INDEX(s) \*/)

План выполнения запроса будет:



Подключим индекс PROMO\_PK.

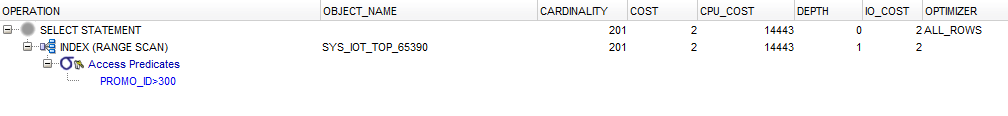
1. SELECT /\*+ INDEX(promotions) \*/ \* FROM SHpromotions WHERE promo\_id > 300;



1. **Что получили? (сравнить оба плана)**

Теперь воспользуемся таблицей, организованной по индексу.

SELECT \* FROM promotions\_iot WHERE promo\_id > 300;



1. **Что получили? (сравнить с предыдущими планами, какой план лучше?)**

## Сканирование с пропуском

Создадим тестовую таблицу и заполним ее.

create table t(c number, d number);

begin

for i in 1..10000 loop

insert into t values(1,i);

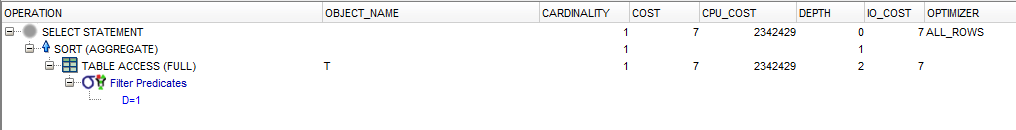
end loop;

end;

create index it on t(c,d);

Создадим простой запрос и сформируем план его выполнения

select count(\*) from t where d=1;

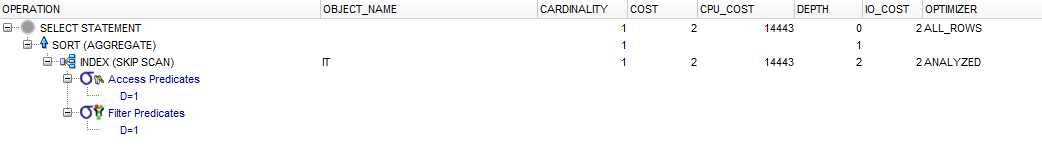


1. **Что получили? (Почему не используется индекс?)**

Сформируем статистику по этой таблице.

execute dbms\_stats.gather\_table\_stats('TEST\_1','T',cascade=>TRUE);

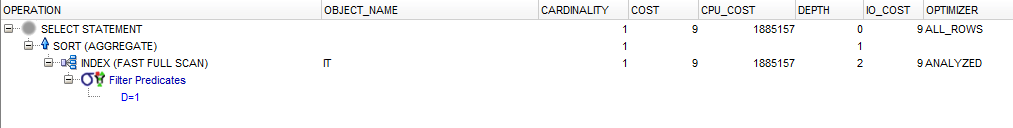
и повторим наш запрос.



1. **Что получили?**

Изменим запрос и укажем, что необходимо использовать индекс, как он есть.

select /\*+ INDEX\_FFS(t it) \*/ count(\*) from t where d=1;



1. **Что получили? (Сравнить использование индексов)**