

Understand Oracle Access Paths

by Lev Moltyaner

Table of Contents

1. Driving Table and Cost versus Rule Based	2
2. Reducing Access to the Same Table in Complex Queries	5
3. Composite Index vs Multiple Single Column Indexes	12
4. UNION vs UNION ALL	16
5. Utilizing Maximum Number of Leading Columns of a Composite Index	22
6. Hints	24
7. Hash Cluster	28

1. Driving Table and Cost versus Rule Based

Request : Select all contracts for a specified gas day from Current Daily Detail table only if they appear on a specific nom in the Issued Nom table.

Query : Use sub-query to get a list of contracts from Issued Nom table, select these contracts from Current Daily Detail table for the specified gas day.

Indexes :

1. Issued Nom

INOM_PK (issued_nom_id, nom_id, ...)

2. Current Daily Detail

UK1_NOM_GAS_DAY_TEMPLATE (gas_day, nom_id, ...)

GAS_DAY_CONTRACT (gas_day, contract_id)

Optimizer :

Will use a NESTED LOOP JOIN to join Issued_Nom rows with Current_Daily_Detail rows (use data from the driving (outer) table to access the inner table).

1. Rule Based :

Selects Current Daily Detail as the driving table :

- Access to both tables is available via INDEX RANGE SCAN on composite indexes
- In both cases only the first column of the index is used
- Main query table is chosen
- Benchmark : 58 sec

2. Cost Based :

- Used Issued_Nom table based on cost estimates
- Benchmark : 2 sec

Query

```
select cdd.contract_id
from energy_volume_uom uom,
service_offering so,
current_daily_detail cdd
/* get all contracted services for specific nom to be issued */
where (cdd.contract_id,
       cdd.service_type,
       cdd.rate_class,
       cdd.service_class,
       cdd.offered_by_party_id) in
      (select
        inom.contract_id,
        inom.service_type,
        inom.rate_class,
        inom.service_class,
        inom.offered_by_party_id
      from issued_nom inom
      where inom.issued_nom_id = 20414
            and inom.from_last_nom_flag = 'N'
            and inom.contract_id is not null )
/* only for relevant gas day */
and cdd.gas_day = '07-jul-95'
and cdd.cdd_type = decode('N','Y',cdd.cdd_type, 'NOM')
/* can't match diversion with a segment entitlement, so ignore for now */
and cdd.service_sub_type <> 'DIV'
/* ignore superceded noms */
and nvl(cdd.nom_status, 'adj') <> 'EXPIRED'
/* join to service offering to get eba_flag */
and so.service_type = cdd.service_type
and so.rate_class = cdd.rate_class
and so.service_class = cdd.service_class
and so.offered_by_party_id = cdd.offered_by_party_id
/* join to uom to get tolerance */
and uom.energy_volume_uom = cdd.energy_volume_uom
group by
  cdd.contract_id,
  cdd.service_type,
  cdd.rate_class,
  cdd.service_class,
  cdd.offered_by_party_id,
  cdd.service_usage,
  so.eba_flag,
  uom.rounding_tolerance,
  cdd.from_trading_location_id,
  cdd.to_trading_location_id,
  cdd.service_sub_type;
```

Rule Based Access Path

SELECT STATEMENT
SORT GROUP BY
FILTER
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN UK1_NOM_GAS_DAY_TEMPLATE
TABLE ACCESS BY ROWID SERVICE_OFFERING
INDEX UNIQUE SCAN SOFF_PK
TABLE ACCESS BY ROWID ENERGY_VOLUME_UOM
INDEX UNIQUE SCAN EVU_PK
TABLE ACCESS BY ROWID ISSUED_NOM
INDEX RANGE SCAN INOM_PK

Cost Based Access Path

SELECT STATEMENT
SORT GROUP BY
FILTER
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
TABLE ACCESS BY ROWID ISSUED_NOM
INDEX RANGE SCAN INOM_PK
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN GAS_DAY_CONTRACT
TABLE ACCESS FULL ENERGY_VOLUME_UOM
TABLE ACCESS FULL SERVICE_OFFERING

2. Reducing Access to the Same Table in Complex Queries

Request : Select all details for a specified gas day from Current Daily Detail table only if they fall into one of four special cases. For each detail get data from the Nom table.

Query : An OR clause dealing with each special case. A common gas_day clause.

Indexes :

1. Nom

NOM_PK (nom_id)

2. Current Daily Detail

FROM_LINK (from_nom_group_number, gas_day, from_trading_location_id)

TO_LINK (to_nom_group_number, gas_day, to_trading_location_id)

Optimizer :

Will use a NESTED LOOP JOIN to join Nom rows with Current_Daily_Detail rows (use data from the driving (outer) table to access the inner table).

1. Consider each case individually

For cases where it is possible to have Nom table as the driving table, use it, other cases, leave Current Daily Detail as the driving table

- Each case will require separate access to Current Daily Detail
- Benchmark : 2 min 15 sec

2. Consider all cases together

- Not possible to have Nom as the driving table in all cases
- Make Current Daily Detail the driving table in all cases
- Benchmark : 33 sec

Query Considering Each OR Case Individually

```
select 1
from cdd_103m3 cdd
, nom
where((nom.nom_id = 20414
and cdd.nom_id = nom.nom_id
and cdd.gas_day = nom.start_gas_day
and cdd.nom_group_number is not null)
or (cdd.supplier_nom_group_number = 26
and to_char(cdd.start_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.start_hours_into_gas_day,'09')) <= '1995070700'
and to_char(cdd.end_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.end_hours_into_gas_day,'09')) > '1995070700'
and '07-jul-95' between cdd.start_gas_day and cdd.end_gas_day
and nom.nom_id = cdd.nom_id
and cdd.internal_external = 'I')
or (nom.nom_id <> 20414
and cdd.nom_id = nom.nom_id
and cdd.nom_group_number = 26
and to_char(cdd.start_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.start_hours_into_gas_day,'09')) <= '1995070700'
and to_char(cdd.end_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.end_hours_into_gas_day,'09')) > '1995070700'
and '07-jul-95' between cdd.start_gas_day and cdd.end_gas_day)
or (nom.nom_id <> 20414
and cdd.nom_id = nom.nom_id
and to_char(cdd.start_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.start_hours_into_gas_day,'09')) <= '1995070700'
and to_char(cdd.end_gas_day,'YYYYMMDD')
|| ltrim(to_char(cdd.end_hours_into_gas_day,'09')) > '1995070700'
and '07-jul-95' between cdd.start_gas_day and cdd.end_gas_day)
and exists
(select 1
from current_daily_detail cdd1
where (nom_id = 20414
or cdd1.supplier_nom_group_number = 26)
and nom_status <> 'EXPIRED'
and gas_day = '07-jul-95'
and cdd1.nom_group_number is not null
and ((cdd1.from_trading_location_id = cdd.location_id
and cdd1.from_nom_group_number = cdd.nom_group_number
and nvl(cdd1.from_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1))
or
(cdd1.to_trading_location_id = cdd.location_id
and cdd1.to_nom_group_number = cdd.nom_group_number
and nvl(cdd1.to_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1)))
)))
```

```
and cdd.gas_day = '07-jul-95'  
group by location  
      ,cdd.nom_group_number  
      ,cdd.nom_group_stream  
having abs(sum(round(decode(cdd.service_class  
      , 'PF', nom_rate * decode(cdd.service_type, 'TRANS', 1, 1)  
      , cdd.nom_rate), 1))) > 0;
```

Access Path for Query Considering Each OR Case Individually

SELECT STATEMENT

FILTER

SORT GROUP BY

CONCATENATION

FILTER

NESTED LOOPS

VIEW CDD_103M3

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

CONCATENATION

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN FROM_LINK

FILTER

NESTED LOOPS

VIEW CDD_103M3

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

FILTER

NESTED LOOPS
VIEW CDD_103M3
PROJECTION
UNION-ALL
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK
TABLE ACCESS BY ROWID NOM
INDEX UNIQUE SCAN NOM_PK
FILTER
NESTED LOOPS
TABLE ACCESS BY ROWID NOM
INDEX UNIQUE SCAN NOM_PK
FILTER
FILTER
VIEW CDD_103M3
PROJECTION
UNION-ALL
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK

Query Considering All OR Cases as One

```
select 1
from cdd_103m3 cdd
    ,nom
where((cdd.nom_id = 20414
    and cdd.nom_group_number is not null)
or (cdd.supplier_nom_group_number = 26
    and cdd.internal_external = 'I')
or (nom.nom_id <> 20414
    and cdd.nom_group_number = 26)
or (nom.nom_id <> 20414
    and exists
        (select 1
        from current_daily_detail cdd1
        where (nom_id = 20414
            or cdd1.supplier_nom_group_number = 26)
            and nom_status <> 'EXPIRED'
            and gas_day = '07-jul-95'
            and cdd.nom_group_number is not null
            and ((cdd1.from_trading_location_id = cdd.location_id
                and cdd1.from_nom_group_number = cdd.nom_group_number
                and nvl(cdd1.from_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1))
            or
                (cdd1.to_trading_location_id = cdd.location_id
                and cdd1.to_nom_group_number = cdd.nom_group_number
                and nvl(cdd1.to_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1)))
        )))
    and cdd.gas_day = '07-jul-95'
    and nom.nom_id = cdd.nom_id
    and to_char(nom.start_gas_day,'YYYYMMDD')
        || ltrim(to_char(nom.start_hours_into_gas_day,'09')) <= '1995070700'
    and to_char(nom.end_gas_day,'YYYYMMDD')
        || ltrim(to_char(nom.end_hours_into_gas_day,'09')) > '1995070700'
    and '07-jul-95' between nom.start_gas_day and nom.end_gas_day
group by location
    ,cdd.nom_group_number
    ,cdd.nom_group_stream
-- balance within rounding tolerance
having abs(sum(round(decode(cdd.service_class
    , 'PF', cdd.nom_rate * decode(cdd.service_type, 'TRANS', 1, 1)
    , nom_rate), 1))) > 0;
```

Access Path for Query Considering All OR Cases as One

SELECT STATEMENT

FILTER

SORT GROUP BY

FILTER

NESTED LOOPS

VIEW CDD_103M3

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

CONCATENATION

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN FROM_LINK

3. Composite Index vs Multiple Single Column Indexes

Request : Select all details for a specified gas day from Current Daily Detail table only if they fall into one of four special cases. For each detail get data from the Nom table.

Query : An OR clause dealing with each special case. A common gas_day clause.

Optimizer :

1. Current Daily Detail Indexes :

FROM_LINK (from_nom_group_number, gas_day, from_trading_location_id)
TO_LINK (to_nom_group_number, gas_day, to_trading_location_id)

- Using FROM_LINK and TO_LINK composite indexes to access Current_Daily_Detail rows
- Benchmark : 33 sec

2. Current Daily Detail Indexes :

GAS_DAY (gas_day)
FROM_NGN (from_nom_group_number)
FROM_TL (from_trading_location_id)
TO_NGN (to_nom_group_number)
TO_TL (to_trading_location_id)

- Using GAS_DAY, FROM_NGN, FROM_TL and GAS_DAY, TO_NGN, TO_TL single column indexes to access Current_Daily_Detail rows
- Benchmark : 4 min

Query

```
select 1
from cdd_103m3 cdd
    ,nom
where((cdd.nom_id = 20414
    and cdd.nom_group_number is not null)
or (cdd.supplier_nom_group_number = 26
    and cdd.internal_external = 'I')
or (nom.nom_id <> 20414
    and cdd.nom_group_number = 26)
-- cdds which touch cdds WITH SNG = in_nom_group (ie. touch WBS)
-- cdds which touch cdds of in_nom
or (nom.nom_id <> 20414
    and exists
        (select 1
        from current_daily_detail cdd1
        where (nom_id = 20414
            or cdd1.supplier_nom_group_number = 26)
            and nom_status <> 'EXPIRED'
            and gas_day = '07-jul-95'
            and cdd.nom_group_number is not null
            and ((cdd1.from_trading_location_id = cdd.location_id
                and cdd1.from_nom_group_number = cdd.nom_group_number
                and nvl(cdd1.from_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1))
            or
                (cdd1.to_trading_location_id = cdd.location_id
                and cdd1.to_nom_group_number = cdd.nom_group_number
                and nvl(cdd1.to_nom_group_stream,-1) = nvl(cdd.nom_group_stream,-1)))
        )))
and cdd.gas_day = '07-jul-95'
and nom.nom_id = cdd.nom_id
and to_char(nom.start_gas_day,'YYYYMMDD')
    || ltrim(to_char(nom.start_hours_into_gas_day,'09')) <= '1995070700'
and to_char(nom.end_gas_day,'YYYYMMDD')
    || ltrim(to_char(nom.end_hours_into_gas_day,'09')) > '1995070700'
and '07-jul-95' between nom.start_gas_day and nom.end_gas_day
group by location
    ,cdd.nom_group_number
    ,cdd.nom_group_stream
-- balance within rounding tolerance
having abs(sum(round(decode(cdd.service_class
    , 'PF', cdd.nom_rate * decode(cdd.service_type, 'TRANS', 1, 1)
    , nom_rate), 1))) > 0;
```

Access Path for Composite Indexes

SELECT STATEMENT

FILTER

SORT GROUP BY

FILTER

NESTED LOOPS

VIEW CDD_103M3

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

CONCATENATION

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN FROM_LINK

Access Path for Multiple Single Column Indexes

SELECT STATEMENT

FILTER

SORT GROUP BY

FILTER

NESTED LOOPS

VIEW CDD_103M3

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN GAS_DAY

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN GAS_DAY

TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE

INDEX RANGE SCAN EHV_PK

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

CONCATENATION

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

AND-EQUAL

INDEX RANGE SCAN TO_TL

INDEX RANGE SCAN TO_NGN

INDEX RANGE SCAN GAS_DAY

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

AND-EQUAL

INDEX RANGE SCAN FROM_TL

INDEX RANGE SCAN FROM_NGN

INDEX RANGE SCAN GAS_DAY

4. UNION vs UNION ALL

Request : Select cumulative quantity from Entitlement History for a nom with contracts in Issued Nom or Current Daily Detail table for a specified gas day.

Query : Use a UNION with two sub-queries, one for Issued Nom table and the other for Current Daily Detail table. If contract appears in both places only want one row processed.

Optimizer :

1. UNION :

- Requires a SORT UNIQUE operation after doing a UNION-ALL
- Benchmark : 4 sec

2. UNION ALL :

- Does not require SORT UNIQUE, done programatically in a PL/SQL block
- Benchmark : 0.5 sec

Query

```
select
    e.contract_id,
    e.service_type,
    e.rate_class,
    e.service_class,
    e.offered_by_party_id,
    e.trading_location_id,
    eh.cumulative_quantity
from
    entitlement_ba_history eh,
    entitlement_ba e
-- CASE 1 : ISSUED NOM
-- must do sub-query because need DISTINCT CS
where 20414 is not null
and (e.contract_id,
     e.service_type,
     e.rate_class,
     e.service_class,
     e.offered_by_party_id) in
(select
    inom.contract_id,
    inom.service_type,
    inom.rate_class,
    inom.service_class,
    inom.offered_by_party_id
from issued_nom inom
where inom.issued_nom_id = 20414
and 20414 is not null
and inom.from_last_nom_flag = 'N'
and inom.contract_id is not null )
and eh.eba_id = e.eba_id
-- only for relevant gas day
and eh.gas_day = '01-jul-95'
and eh.gas_day <> e.start_date - 1
and not exists
(select 1
from cs_detail_entitlement ent
where ent.eba_id = e.eba_id
and '01-jul-95' between ent.start_date and ent.end_date
and eh.cumulative_quantity between nvl(ent.lower_bound,-99999999.9)
and nvl(ent.upper_bound,99999999.9));
```

UNION

select

e.contract_id,
e.service_type,
e.rate_class,
e.service_class,
e.offered_by_party_id,
e.trading_location_id,
eh.cumulative_quantity

from

entitlement_ba_history eh,
entitlement_ba e

-- RECEIVED NOM

-- 1. must do received nom, even if it is actually an issued nom because
-- may not have rows in issued_nom table yet
-- 2. need a subquery to cdd to get all contracts on that cdd then use
-- these contracts on cdd to get all cdds for gas day (otherwise
-- will miss partial day cdds)

where 20414 is not null

and (e.contract_id,

e.service_type,
e.rate_class,

e.service_class,
e.offered_by_party_id) in

(select

cdd2.contract_id,
cdd2.service_type,
cdd2.rate_class,
cdd2.service_class,
cdd2.offered_by_party_id
from current_daily_detail cdd2
where cdd2.nom_id = 20414
and 20414 is not null
and cdd2.gas_day = '01-jul-95'

and cdd2.contract_id is not null)

and eh.eba_id = e.eba_id

-- only for relevant gas day

and eh.gas_day = '01-jul-95'

and eh.gas_day <> e.start_date - 1

and not exists

(select 1

from cs_detail_entitlement ent

where ent.eba_id = e.eba_id

and '01-jul-95' between ent.start_date and ent.end_date

and eh.cumulative_quantity between nvl(ent.lower_bound,-99999999.9)

and nvl(ent.upper_bound,99999999.9));

Access path with UNION operator

SELECT STATEMENT

PROJECTION

SORT UNIQUE

UNION-ALL

FILTER

FILTER

NESTED LOOPS

NESTED LOOPS

TABLE ACCESS BY ROWID ENTITLEMENT_BA_HISTORY

INDEX RANGE SCAN EBAH_PK

TABLE ACCESS BY ROWID ENTITLEMENT_BA

INDEX UNIQUE SCAN EBA_PK

TABLE ACCESS BY ROWID ISSUED_NOM

INDEX RANGE SCAN ISSUED_NOM_NOM

TABLE ACCESS FULL CS_DETAIL_ENTITLEMENT

FILTER

FILTER

NESTED LOOPS

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN NOM_ID_GAS_DAY

TABLE ACCESS BY ROWID ENTITLEMENT_BA

INDEX RANGE SCAN UK1_EBA

TABLE ACCESS BY ROWID ENTITLEMENT_BA_HISTORY

INDEX RANGE SCAN EBAH_PK

TABLE ACCESS FULL CS_DETAIL_ENTITLEMENT

PL/SQL Block to Handle Duplicate Rows from Query with UNION ALL operator

```
DECLARE
```

```
cursor eba_detail_cur is
```

```
    ... identical cursor, but replace UNION with UNION ALL ...
```

```
save_eba_det eba_detail_cur%ROWTYPE;
```

```
BEGIN
```

```
    for eba_detail in eba_detail_cur(
```

```
        in_gas_day, in_operator_id, in_contract_id, in_nom_id) loop
```

```
        if( eba_detail_cur%ROWCOUNT <> 1
```

```
            and eba_detail.contract_id = save_eba_det.contract_id
```

```
            and eba_detail.service_type = save_eba_det.service_type
```

```
            and eba_detail.rate_class = save_eba_det.rate_class
```

```
            and eba_detail.service_class = save_eba_det.service_class
```

```
            and eba_detail.offered_by_party_id = save_eba_det.offered_by_party_id
```

```
            and eba_detail.trading_location_id = save_eba_det.trading_location_id
```

```
            and eba_detail.cumulative_quantity = save_eba_det.cumulative_quantity ) then
```

```
                null;
```

```
        else
```

```
            ....
```

```
        end if;
```

```
        save_eba_det.contract_id := eba_detail.contract_id;
```

```
        save_eba_det.service_type := eba_detail.service_type;
```

```
        save_eba_det.rate_class := eba_detail.rate_class;
```

```
        save_eba_det.service_class := eba_detail.service_class;
```

```
        save_eba_det.offered_by_party_id := eba_detail.offered_by_party_id;
```

```
        save_eba_det.trading_location_id := eba_detail.trading_location_id;
```

```
        save_eba_det.cumulative_quantity := eba_detail.cumulative_quantity;
```

```
    end loop;
```

Access path with UNION ALL operator

SELECT STATEMENT

PROJECTION

UNION-ALL

FILTER

NESTED LOOPS

NESTED LOOPS

VIEW

SORT UNIQUE

FILTER

TABLE ACCESS BY ROWID ISSUED_NOM

INDEX RANGE SCAN ISSUED_NOM_NOM

TABLE ACCESS FULL ENTITLEMENT_BA

TABLE ACCESS BY ROWID ENTITLEMENT_BA_HISTORY

INDEX RANGE SCAN EBAH_PK

TABLE ACCESS BY ROWID CS_DETAIL_ENTITLEMENT

INDEX RANGE SCAN EBA_ID

FILTER

NESTED LOOPS

NESTED LOOPS

VIEW

SORT UNIQUE

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN NOM_ID_GAS_DAY

TABLE ACCESS FULL ENTITLEMENT_BA

TABLE ACCESS BY ROWID ENTITLEMENT_BA_HISTORY

INDEX RANGE SCAN EBAH_PK

TABLE ACCESS BY ROWID CS_DETAIL_ENTITLEMENT

INDEX RANGE SCAN EBA_ID

5. Utilizing Maximum Number of Leading Columns of a Composite Index

Request : Select all nom details which link to a specified nom group and which are on a nom that falls on the specified gas day.

Query : Select specified nom group from Current Daily Detail and join to Nom table to check if these details belong to a nom with dates which cover the specified gas day.

Index :

1. Current Daily Detail Indexes :

TO_LINK (to_nom_group_number, gas_day, to_trading_location_id)

Optimizer :

1. Using only the first column of a three-column composite index
 - Do not know the gas day when accessing Current Daily Detail
 - Benchmark : 48 sec
2. Using the first two columns of the three-column composite index
 - Explicitly state the gas day for Current Daily Detail
 - Benchmark : 4 sec

Query using only the first column of the TO_LINK composite index

```
select 1
from current_daily_detail cdd
,estimated_heat_value ehv
,nom
where cdd.to_nom_group_number = 26
and cdd.nom_id <> 20414
and nom.start_gas_day = cdd.gas_day
and nom.nom_id = cdd.nom_id
and to_char(nom.start_gas_day,'YYYYMMDD')
|| ltrim(to_char(nom.start_hours_into_gas_day,'09')) <= '1995070702'
and to_char(nom.end_gas_day,'YYYYMMDD')
|| ltrim(to_char(nom.end_hours_into_gas_day,'09')) > '1995070702'
and '07-jul-95' between nom.start_gas_day and nom.end_gas_day
and cdd.gas_day between ehv.start_date and ehv.end_date
and cdd.from_trading_location_id = ehv.trading_location_id
and not exists
(select 1
from issued_nom
where issued_nom_id = 20414
and daily_detail_id = cdd.daily_detail_id
and category = 'UPSTREAM');
```

Query using only the first column of the TO_LINK composite index

```
select 1
from current_daily_detail cdd
,estimated_heat_value ehv
,nom
where cdd.to_nom_group_number = 26
and cdd.nom_id <> 20414
and cdd.gas_day = '07-jul-95'
-- and nom.start_gas_day = cdd.gas_day
and nom.nom_id = cdd.nom_id
and to_char(nom.start_gas_day,'YYYYMMDD')
|| ltrim(to_char(nom.start_hours_into_gas_day,'09')) <= '1995070702'
and to_char(nom.end_gas_day,'YYYYMMDD')
|| ltrim(to_char(nom.end_hours_into_gas_day,'09')) > '1995070702'
and '07-jul-95' between nom.start_gas_day and nom.end_gas_day
and cdd.gas_day between ehv.start_date and ehv.end_date
and cdd.from_trading_location_id = ehv.trading_location_id
and not exists
(select 1
from issued_nom
where issued_nom_id = 20414
and daily_detail_id = cdd.daily_detail_id
and category = 'UPSTREAM');
```

Access Path for Both Queries

```
SELECT STATEMENT
  FILTER
    NESTED LOOPS
      NESTED LOOPS
        TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
          INDEX RANGE SCAN TO_LINK
        TABLE ACCESS BY ROWID NOM
          INDEX UNIQUE SCAN NOM_PK
      TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
        INDEX RANGE SCAN EHV_PK
    INDEX UNIQUE SCAN INOM_PK
```

Note : No table access to Issued Nom table, all required information is located in the INOM_PK index

6. Hints

Request : Select current daily details for contracted services of nom groups of a specific operator.

Query : Select all contracted services with nom group numbers belonging to the specified operator.
Use these contracted services to select from Current Daily Detail.

Optimizer :

1. Rule Based

- Uses Current Daily Detail as the driving table
- Uses data from the driving table to join to remaining tables using indexes
- Benchmark : 5 sec

2. Cost Based without Hints

- Using the Current Daily Detail as the driving table
- Uses data from the driving table to join to remaining tables uses full table scans
- Benchmark : 4 sec

3. Cost Based with Hints

- Forcing the driving table to be Nom Group and Contracted Service
- Using Contracted Service for a more efficient index on Current Daily Detail
- Benchmark : 3 sec

Query Without Hints

```
select  cdd.contract_id,
        sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'N', 1, 0)) normal_quantity,
        sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'D', 1, 0)) discretionary_quantity,
        sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'O', 1, 0)) overrun_quantity
from    nom_group ng,
        contracted_service cs,
        current_daily_detail cdd,
        service_offering so,
        energy_volume_uom uom
where   ng.operator_party_id_for_hour = 1
and     cs.nom_group_number = ng.nom_group_number
and     cdd.contract_id = cs.contract_id
        and cdd.service_type = cs.service_type
        and cdd.rate_class = cs.rate_class
and     cdd.service_class = cs.service_class
and     cdd.offered_by_party_id = cs.offered_by_party_id
and     cdd.gas_day = '03-aug-95'
        and so.service_type = cdd.service_type
        and so.rate_class = cdd.rate_class
        and so.service_class = cdd.service_class
        and so.offered_by_party_id = cdd.offered_by_party_id
and     uom.energy_volume_uom = cdd.energy_volume_uom
group  by cdd.contract_id;
```

Access Path for Rule Based Optimization

SELECT STATEMENT
SORT GROUP BY
FILTER
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ENERGY_VOLUME_UOM
INDEX UNIQUE SCAN EVU_PK
TABLE ACCESS BY ROWID SERVICE_OFFERING
INDEX UNIQUE SCAN SOFF_PK
TABLE ACCESS BY ROWID CONTRACTED_SERVICE
INDEX UNIQUE SCAN CSER_PK
TABLE ACCESS BY ROWID NOM_GROUP
INDEX UNIQUE SCAN NGRO_PK

Access Path for Cost Based Without Hints

SELECT STATEMENT
SORT GROUP BY
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS FULL SERVICE_OFFERING
TABLE ACCESS FULL CONTRACTED_SERVICE
TABLE ACCESS FULL NOM_GROUP
TABLE ACCESS FULL ENERGY_VOLUME_UOM

Query with Hints

```
select /*+ ordered use_nl(cs) index(cs nom_group_number) use_nl(cdd) */
cdd.contract_id,
sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'N', 1, 0)) normal_quantity,
sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'D', 1, 0)) discretionary_quantity,
sum(cdd.nom_quantity * Decode(cdd.overrun_indicator, 'O', 1, 0)) overrun_quantity
from nom_group ng,
contracted_service cs,
current_daily_detail cdd,
service_offering so,
energy_volume_uom uom
where ng.operator_party_id_for_hour = 1
and cs.nom_group_number = ng.nom_group_number
and cdd.contract_id = cs.contract_id
and cdd.service_type = cs.service_type
and cdd.rate_class = cs.rate_class
and cdd.service_class = cs.service_class
and cdd.offered_by_party_id = cs.offered_by_party_id
and cdd.gas_day = '03-aug-95'
and so.service_type = cdd.service_type
and so.rate_class = cdd.rate_class
and so.service_class = cdd.service_class
and so.offered_by_party_id = cdd.offered_by_party_id
and uom.energy_volume_uom = cdd.energy_volume_uom
group by cdd.contract_id;
```

Access Path for Cost Based Optimizer With Hints

```
SELECT STATEMENT
SORT GROUP BY
FILTER
  NESTED LOOPS
    NESTED LOOPS
      NESTED LOOPS
        NESTED LOOPS
          TABLE ACCESS FULL NOM_GROUP
          TABLE ACCESS BY ROWID CONTRACTED_SERVICE
          INDEX RANGE SCAN NOM_GROUP_NUMBER
          TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
          INDEX RANGE SCAN CONTRACT_GAS_DAY
          TABLE ACCESS FULL SERVICE_OFFERING
          TABLE ACCESS FULL ENERGY_VOLUME_UOM
```

7. Hash Cluster

Request : Check if related current daily details balance to zero.

Query : Select from cdd_uom view and nom. The view is a union of two queries which join current daily detail to estimated value to perform unit of measure conversion.

Optimizer :

For each current daily detail accesses estimated heat value table.

1. Estimated heat value table is not hashed
 - Uses a RANGE SCAN on estimated heat value table index
 - Benchmark : 1 min 48 sec
2. Change estimated heat value table to a hash cluster
 - Hash values are based on trading_location_id
 - Benchmark : 47 sec

Query Based On cdd_uom View

```
select 1
from cdd_uom cdd
, nom
where((cdd.nom_id = 30921
and cdd.nom_group_number is not null)
or (cdd.supplier_nom_group_number = 26
and internal_external = 'I')
or (cdd.nom_idu <> 30921
and cdd.nom_group_number = 26)
or (cdd.nom_id <> 30921
and exists
(select 1
from current_daily_detail cdd1
where (cdd1.nom_id = 30921
or cdd1.supplier_nom_group_number = 26)
and cdd1.nom_status <> 'EXPIRED'
and cdd1.gas_day = '29-nov-95'
and cdd1.nom_group_number is not null
and (( cdd1.from_trading_location_id = cdd.location_id
and cdd1.from_nom_group_number = cdd.nom_group_number
and nvl(cdd1.from_nom_group_stream,-1)
= nvl(cdd.nom_group_stream,-1))
or
( cdd1.to_trading_location_id = cdd.location_id
and cdd1.to_nom_group_number = cdd.nom_group_number
and nvl(cdd1.to_nom_group_stream,-1)
= nvl(cdd.nom_group_stream,-1))))))
and cdd.gas_day = '29-nov-95'
and nom.nom_id = cdd.nom_id
and to_char(start_gas_day,'YYYYMMDD')
|| ltrim(to_char(start_hours_into_gas_day,'09')) <= '1995112908'
and to_char(end_gas_day,'YYYYMMDD')
|| ltrim(to_char(end_hours_into_gas_day,'09')) > '1995112908'
and '29-nov-95' between start_gas_day and end_gas_day
group by location
,cdd.nom_group_number
,cdd.nom_group_stream
having abs( sum( round(cdd.nom_rate
*decode(service_class
,'PF', decode(service_type,'TRANS',1,1),1), 1))) > 1)
```

Create hash cluster

```
create cluster ehv_hash_clus  
(trading_location_id number(5))  
hashkeys 120  
hash is trading_location_id  
storage (initial 550K next 200K pctincrease 0)  
pctfree 0;
```

```
rename estimated_heat_value to ehv;  
create table estimated_heat_value  
cluster ehv_hash_clus(trading_location_id)  
as select * from ehv;
```

```
drop table ehv;
```

Access Path not using a Hash Cluster

FILTER
TABLE ACCESS FULL DUAL
FILTER
SORT GROUP BY
FILTER
NESTED LOOPS
VIEW CDD_UOM
PROJECTION
UNION-ALL
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK
NESTED LOOPS
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
TABLE ACCESS BY ROWID ESTIMATED_HEAT_VALUE
INDEX RANGE SCAN EHV_PK
TABLE ACCESS BY ROWID NOM
INDEX UNIQUE SCAN NOM_PK
CONCATENATION
FILTER
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN TO_LINK
FILTER
TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL
INDEX RANGE SCAN FROM_LINK

Access Path using a Hash Cluster

FILTER

TABLE ACCESS FULL DUAL

FILTER

SORT GROUP BY

FILTER

NESTED LOOPS

VIEW CDD_UOM

PROJECTION

UNION-ALL

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS HASH ESTIMATED_HEAT_VALUE

NESTED LOOPS

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

TABLE ACCESS HASH ESTIMATED_HEAT_VALUE

TABLE ACCESS BY ROWID NOM

INDEX UNIQUE SCAN NOM_PK

CONCATENATION

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN TO_LINK

FILTER

TABLE ACCESS BY ROWID CURRENT_DAILY_DETAIL

INDEX RANGE SCAN FROM_LINK