**Inline Views or Derived Tables or Sub Query in From Clause**

Whenever derived columns are going to be required again & again in the output with some different results, then repeating the same expression again & again will be tedious & re parsing effort will be there.

**Example 1:**

selectename,

sal,

comm,

sal+NVL(comm,0.30\*Sal)asTotal,

(sal+NVL(comm,0.30\*Sal))\*0.5as"50% Less",

(sal+NVL(comm,0.30\*Sal))\*0.4as"40% Less",

(sal+NVL (comm,0.30\*Sal))\*0.3as"30% Less"

fromemp;

The expression (sal+NVL(comm,0.30\*Sal)) is getting repeated again & again, (not good)

Instead of this, the expression can be defined **once** in the derived table & then further can be used **as a column itself** **in the main select statement in the column list, or where clause or group by clause or having clause !!!!**

selectename,sal,comm, **total**,

**total** \*0.5as"50% Less",

**total** \*0.4as"40% less",

**total** \*0.30as"30% Less"

from (selectename,sal,comm,

**sal + NVL(comm, 0.30 \* Sal) as total**

fromemp

) **e**;

Also the derived column total can be used in the **where clause as it is** of the main select statement!!!!!!

select ename, sal, comm, **total**

from **(**select ename, sal, comm, **sal + NVL(comm, 0.30 \* Sal) as total** from emp**)**

where **total** > 2000;

**Example 2:**

|  |
| --- |
| selectename,sal,Bonus,sal+Bonusas"Total Extra",  Bonus+2000as"Arrears", (sal+Bonus)\*0.02asTax  from (selectename,sal,  casewhensal>=5000then (0.10\*sal)  whensal>=3000then (0.20\*sal)  whensal>=2000then (0.30\*sal)  whensal>=1000then (0.40\*sal)  else  (0.50\*sal)  endasBonus  fromemp)e; |

**Example 3:**

**--Derived Column logic getting repeated in Column List & Group By clause**

Select Case

When to\_char(Hiredate,'yyyy') = 1980 then 'In First Year'

When to\_char(Hiredate,'yyyy') = 1981 then 'In Second Year'

When to\_char(Hiredate,'yyyy') = 1983 then 'In Third Year'

Else

'Above third year'

End as "Year of Joining",

Sum(Sal) as Total

from emp

Group by Case

When to\_char(Hiredate,'yyyy') = 1980 then 'In First Year'

When to\_char(Hiredate,'yyyy') = 1981 then 'In Second Year'

When to\_char(Hiredate,'yyyy') = 1983 then 'In Third Year'

Else

'Above third year'

End

**-- Writing the Case function in the derived table to eliminate redundancy**

Select "Year of Joining", Sum(Sal) as Total

from **(Select Case**

**When to\_char(Hiredate,'yyyy') = 1980 then 'In First Year'**

**When to\_char(Hiredate,'yyyy') = 1981 then 'In Second Year'**

**When to\_char(Hiredate,'yyyy') = 1983 then 'In Third Year'**

**Else**

**'Above third year'**

**End as "Year of Joining",**

**Sal**

**From Emp**

**)**

Group By "Year of Joining";

**Valid Column Combination Hints**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Combination** | **Technique to be used** | **Example** |
| 1 | AC + GC | Group By | Select deptno,max(sal) from emp  Group by deptno |
| 2 | AC + NGC | Sub Query, i.e Simple Sub Query or Non-Correlated SQ | Select ename,sal from emp  Where sal = (select max(sal) from emp) |
| 3 | AC + GC + NGC + Expr | Inline view or Derived Table or From Clause Sub Query |  |

**AC: Aggregate Column (Max(Sal))**

**GC: Groupable Column (Deptno, Job)**

**NGC: Non-groupable column (Empno, Ename)**

**Expr: Expression Column ( Sal - 500 as “Net\_Salary”)**

**To display names, salaries, job, average salary and difference (raise) with average salary of those employees who earn more than the average salary in their own jobs.**

SELECT a.ename, a.sal, a.job, b.SalAvg, (a.sal-b.SalAvg) as "Raise"

FROM emp a INNER JOIN (SELECT job,

Round(AVG(sal),2) as SalAvg

FROM emp

GROUP BY job) b

ON a.job = b.job

WHERE a.sal > b.SalAvg;

**Example 2 –**

Display the name,sal, deptno, highest sal of that deptno and difference of employees who are earning sal less than the highest of their own deptno

**Example 3 –**

To see employees who earn salary less than the average salary **of their own job.**

Another area where sub query in from clause can be used is for performing **Top-N Analysis**

1. **To display the fourth highest salary**

select min(sal) as Fourth\_Highest

from **(select distinct sal from emp order by sal desc) abc**

where rownum <=4;

**Fourth\_Highest**

**----------**

**2850**

**2) To display the fourth lowest salary**

select max(sal)

from **(select distinct sal from emp order by sal asc)**

where rownum <=4

**MAX(SAL)**

**----------**

**1250**

3) **To display record (s) of the 4th highest salary value.**

|  |
| --- |
| select \* from emp  where sal = (select min(sal)  from (select distinct sal  from emp order by sal desc  )  where rownum <=4  ); |

**4) To display the records of top 3 highest salary employees:**

Select \*

from (select \* from emp order by sal desc)

where rownum <= 4; -- Not always effective

OR

**Select a.\***

**from emp a join (select \* from (select distinct sal from emp order by sal desc) abc**

**where rownum <=3**

**) b**

**On a.sal = b.sal; -- More Effective**