

## HAVING clause

→ It works

→ Having clause works as filter  
on top of grouped rows returned by  
group by clause.

→ This clause cannot be replaced  
as where clause because  
where clause is applied at  
row level and it is applied at  
group level.

→ It is always used with  
group by clause.

Eg:-

Select department name, sum(Sal)  
from Emp  
group By department name;

Department Name	sum(Salary)
Finance	50,000
HR	20,000
Production	30,000
Sales	10,000

Now if i want to use having clause,

Eg  
Select department name , sum(salary)  
from employee  
group by department name  
having sum(salary)  $\geq 30,000$  ;

<u>Department Name</u>	<u>sum(salary)</u>
Finance	50,000
Production	30,000

100

## HAVING CLAUSE :-

→ Having clause always used along with group by clause

→ following all the rules that need to be followed for using the group by and having clause

\* columns listed in the select statement have to be listed in group by clause.

\* only group functions can be used in the having clause.

Fq:-

Select job, count(\*) from emp where job in ('clerk', 'sales man') group by job having count(\*) >= 3 order by job

## COPYING THE DATA from Source to Target :-

→ Copying a data from source to target.

Syntax of copying the data from source to target :-

Create table <target table Name> as  
Select \* from <source table Name>;

Eg:-

Create table target as select \* from xyz;

Copying a table from one table to  
another without data :-

Create table <target table name> as select  
\* from <source table name> where  
<false condition>

Eg:- Create table target as select  
\* from xyz where 1=9;

Copying Specific column data from source to target :-

Create table <target table Name> as  
Select col1, col2 ... coln from  
<source table name>;

Eg:- Create table target 2 as select  
eid, ename from xyz;

Copying a table from one table to another with specific column without data

Create table <target Table Name> as  
Select col1, col2.. coln from <source  
table name> where < false condition>

Eg:-  
Create table target 3 as select eid, ename  
from xyz where t=2

Syntax copying a table from one table  
to another table with specific  
records :-

create table <target table name> as  
select \* from <source table name>  
where <condition>

Eg,-

create table target4 as select \* from  
xyz where eid in(101)

Syntax of copying only data from  
source table to target. (not table  
only data) :-

insert into <target table name> select \* from  
<source table name>;

Eg:- insert into target5 select \* from  
xyz;

→ To perform the above  
operation user has to remember  
two important points:-

- i) Number of columns in the source table should equals to number of columns in the target table.
- ii) Data types & sizes of source table should matched with target table data types and sizes.

→ column names can be different doesn't matter.

## MULTI TABLE INSERT STATEMENT

→ the insert - select statement can be used to insert rows into multitable as the part of single . dml statement.

→ Multitable insert statement can be used in datawarehousing systems to transfer data from one or more operational source to set of target tables.

There are 4 types of multiple insert statements :-

- i) unconditional insert all
- ii) conditional insert all
- iii) conditional insert all
- iv) pivoting insert.

## ① unconditional insert all :-

Syntax :-

Eg:- Insert all

into target values (empno, ename, sal,  
deptno)

into target values (empno, ename, sal,  
deptno)

Select empno, ename, sal, deptno from  
emp



→ I have copied the data  
from emp to two target table.

## ③ Conditional insert all :-

Syntax :- Insert all

When deptno = 10 then

into target ~~10~~ values (empno, ename,  
sal, deptno)

When deptno = 20 then

into target ~~1~~ values (empno, ename,  
sal, deptno)

Select empno, ename, sal, deptno  
from emp



Insert first row :-

syntax:-

insert first -

when depthno → 1 then

into target values (empno, ename, sal,  
depthno)

when depthno = 20 then

into target 1 values (empno, ename, sal,  
depthno)

else

into target 3 values (empno, ename,  
sal, depthno)

Select empno, ename, sal, depthno from  
emp;

Pivoting Insert :-

Pivoting is an operation in which you need to build a transformation such that each record from any input stream, such as a non-relational database table, must be converted into multiple records for more relational database table environment.

Syntax :-

Insert all into

Sales\_info Values

(employee\_id, week\_id, sales\_mon)

into Sales\_info values

(employee\_id, week\_id, sales\_TUE)

into Sales\_info values

(employee\_id, week\_id, sales\_WED)

into Sales\_info values

(employee\_id, week\_id, sales\_THUR)

into Sales\_info values

(employee\_id, week\_id, sales\_FRI)

Select employee\_id, week\_id, sales\_mon,

sales\_TUE, sales\_WED, sales\_THUR,

sales\_FRI from sales\_source\_date

→ If you want to show the sum in the table itself not separately then use the following command.

Select \* from Sales\_info  
union

Select null, null, sum(sales) from  
Sales\_info;

## MERGE :- (update + Insert)

→ Using this command you can update or insert a row conditionally into a table, thus avoids multiple update statements. The decision whether to update or insert into the target table is based on the condition in the ON clause. The target statement is suitable in the number of dwh applications.

For eg:- in a dwh application, you need to work with data coming from multiple sources, some of which may be duplicate, with the merge statement you can conditionally add or modify rows.

Syntax :-

Merge into target +  
using emp e  
on (t.empno = e.empno)

when matched then  
update set t.ename = e.ename,  
t.sal = e.sal,  
t.deptno = e.deptno

when not matched then  
insert values (e.empno, e.ename, e.sal,  
e.deptno).

## Case Statement :-

→ It is also like a conditional statement like if, the difference between if condition and case statement is in if condition control need to check all if conditions till the condition satisfied. but whereas in case statement whichever condition all are satisfied then the control automatically jumps to the concerned case-statement and execute the statements. ie why case statement is faster than if condition.

→ In 'if' condition we can check the condition or relation between more than one variable at time but whereas in case statement all the conditions you can apply on only one variable.

Q. WAP to display the increase the employee salary based on the following conditions

i) if the employee job is president then increase 3000 salary

- ii) if the employee job is manager  
then increase 2000 salary
- iii) if the employee job is analyst  
then increase 1500 salary
- iv) if the employee job is clerk  
then increase 1000 salary and  
for others increase 500 salary
- v) for others increase 500 salary

Syntax :-

Select empno, ename, job, sal,

Case job

When 'PRESIDENT' then sal + 3000

When 'MANAGER' then sal + 2000

When 'ANALYST' then sal + 1500

When 'CLERK' then sal + 1000

else

sal + 500

end nested if from emp

/

--- Searched Case Statement ---

Select empno, ename, job, sal,  
Case

when job = 'PRESIDENT' then sal+3000

when job = 'MANAGER' then sal+2000

when job = 'ANALYST' then sal +1500

when job = 'CLERK' then sal +1000

else

sal+500

end new sal from emp)

1

you can do it

# TCL

(Transaction Control Language)

→ This is the 3<sup>rd</sup> sub language in SQL which contains 3 commands :-

- i) commit
- ii) rollback
- iii) save point

Transaction :-

Any operation which we can perform on database by using DML commands such as update, delete & insert is known as transaction.

Session :-

→ It is an interval of time or span of time ie the moment the user login to the database to the moment user logoff from database. The interval of time is known as session.

→ We can terminate the session in two ways :-

- i) normal termination
- ii) abnormal termination

### ① Normal Termination :-

→ Terminating the session by using either by executing exit or quit command is known as Normal termination.

Ex:-      exit/quit.

### ② Abnormal termination :-

→ Terminating the session by directly switch off the CPU button or power off or close the window.

NOTE :- Normal termination always saves our transaction whereas abnormal termination does not save our transaction.

### ① commit :- (Save the data)

→ This command is used to save the transactions permanently from :-

i) The moment the user login to the database to till execute this command

OR

ii) The moment user executing rollback command to till execute this command.

Ex:- Insert into book  
( ' 61no ', ' 61kttitle ', ' 61Author ',  
update book  
set 6 kno = ' 1234 '  
commit ;

→ (undo the transaction)

### (ii) Rollback Statement :-

→ This statement restores the system to its previous state prior to the changes you have made to the database.

Eg:- Insert into book

(`BKNo', `BKTitle', `BKAuthor',  
 `BKPublis') ;

Update Book

Set BKNo = 1224 ;

commit ;

But wait perhaps you made an error you should type 1234, then use Rollback command.

Rollback ;

### (iii) Save point :-

→ It will create system created number for your transaction and for that transaction you can rollback.

→ It is ~~one~~ partial rollback.

→ Instead of doing complete rollback of all the transaction we can do partial rollback.

Syntax :-

Savepoint savepoint name ;

## emp table

select \* from emp ; → show complete table

Now,

Save point e10;

Delete from emp where deptno = 10;

Save point e20;

Delete from emp where deptno = 20;

Savepoint e30;

Delete from emp where dept no = 30;

Select \* from emp ; → empty table.

// Now if we want to rollback;

rollback to e20;

Select \* from emp ; → (only e20 e30  
dept will  
show)

rollback to e10;

Select \* from emp ; → (complete table  
will be there)

# KEYS

→ Key is a single attribute or the set of attributes of relation which can uniquely identify the record within that relation.

Types of keys :-

① Candidate key :-

It is the column or set of columns in the table that qualifies for uniqueness of each row.

- A table can have one or more than one candidate keys.
- It is also known as minimal super key.

Table name :- employee

Employee ID	Employee Name	DOB	DOJ	SSN

Eg:- (Employee ID , SSN)

Table name :- employee I

Registration no.	Employee ID	Name	Salary

Eg:- (ID and Registration no.)

② Primary key :- It is the candidate key attribute/column that is most suited to maintain the uniqueness in the table.

Eg:- Employee ID in employee

OR

\* In entity set employee 1 either Reg. No is primary key or ID is primary key

\* An entity set has only one primary key

③ Alternate keys :-

→ All candidate keys other than primary key are alternate keys.

Eg:- SSN in employee

OR

If you take ID as primary key then Registration No. is alternate key in employee 1

④ Super key :- It is the superset of candidate key.

→ If you add any other column to candidate key it becomes super key.

Eg :-

In employee,  
(employee ID, employee Name,  
(employee ID, Name) DOB)

In employee 1

(ID, Name, salary, Reg No)  
(ID, Name, Reg No)

~~Reg No~~

⑤ Composite key :- If a table do not have a single column that qualifies for a candidate key, then you have to select 2 or more column to make a row unique.

Eg :- If we don't have employee ID then,

✓ (Employee Name + DOB + DOJ)

⑥ Secondary key :-

An attribute or set of attributes which doesn't identify data uniquely is secondary key.

Eg :- Name, salary & Dept No.

⑦ Foreign key :-

employee

Reg. No	ID	Name	Salary	Dept ID
101	101	John	20000	1

Department

Dept ID	Dept - Name
1	IT

→ Foreign key is an attribute in any entity set which is also the primary key in other entity set.

Eg. :- Dept ID  
This is the attribute in employee 2. Also the primary key in department.

Dept ID	Dept Name	Dept ID
1	IT	1
2	HR	2
3	Marketing	3
4	Sales	4

# INTEGRITY CONSTRAINTS

## ① Entity Integrity Rule

→ Primary key in any relation cannot be null.

## ② Referential Integrity Rule :-

→ Foreign key can be either null or it can have only those values which are present in primary key which is related.

EID	Name	salary	Dept-ID
1	Amit	6K	1A
2	Sumit	10K	2C
3	Delit	15K	4F → not allowed
4	Deepak	9K	3F
5	Sandeep	4K	-

Dept ID	Dept Name	null is allowed
1A	Account	
2C	Computer	
3B	Electrical	
4C	Civil.	

## Domain Constraints :-

→ Restrictions which we applied on domain are domain constraints.

EID	Name	Age	Salary
1	Aditya	22	10K
2	Dinesh	-18	5600
3	Sunit	25	8000
4	Lalit	20	ABC
5	Gaurav	23	1,000

## Key constraints :-

→ Duplicate value in primary key is invalid.

## ) Tuple uniqueness constraints :-

→ Duplicate tuples within a single relation is not allowed.