Lesson 2

**Querying Data**

To Create a Database :

create database sha

To Drop a Database

drop database sha

Create table emp(empid int,empname char(20), empaddress varchar(35), empcity char(20), empph int);

insert into emp values (001,'sha','123shagsh', 'cbe', 123456 );

insert into emp values (002,'palani','143shagsh', 'chennai', 456789 );

insert into emp values (003,'ramesh','789shagsh', 'salem', 102030 );

insert into emp values (004,'siva','165shagsh', 'cbe', 506070 );

select \* from emp;

**SELECT**

select \* from emp

**WHERE**

select \* from emp where empname = 'sha'

select \* from emp where empcity = 'cbe'

**DISTINCT Keyword**

select \* from emp;

select distinct \* from emp;

select distinct \* from emp where empcity ='cbe';

select all \* from emp;

TRUNCATE is a DDL command whereas DELETE is a DML command.

2>TRUNCATE is much faster than DELETE.

Reason:When you type DELETE.all the data get copied into the Rollback Tablespace first.then delete operation get performed.Thatswhy when you type ROLLBACK after deleting a table ,you can get back the data(The system get it for you from the Rollback Tablespace).All this process take time.But when you type TRUNCATE,it removes data directly without copying it into the Rollback Tablespace.Thatswhy TRUNCATE is faster.Once you Truncate you cann't get back the data.

3>You cann't rollback in TRUNCATE but in DELETE you can rollback.TRUNCATE removes the record permanently.

4>In case of TRUNCATE ,Trigger doesn't get fired.But in DML commands like DELETE .Trigger get fired.

5>You cann't use conditions(WHERE clause) in TRUNCATE.But in DELETE you can write conditions using WHERE clause

select\*fromstudent

createtableEMP (EIDINTIDENTITY(10,1),ENAMEVARCHAR(20))

DROPTABLEEMP

TRUNCATETABLEEMP

INSERTINTOEMPVALUES('PETER')

--MANUAL INPUT FOR IDENTITY COLUMN

SETIDENTITY\_INSERTEMPON

INSERTINTOEMP (eid,ename)VALUES(10,'kalai')

--inserting Autonumber in Eid column

SETIDENTITY\_INSERTEMPOff

INSERTINTOEMPVALUES('PETER123')

SELECT\*INTOEMPINFOFROMEMP

selectmax(salary)fromemployeegroupby(LOCATION),salaryhavingSALARY>=22000

selectsalaryfromemployeegroupby(LOCATION)

selectmax(salary)fromemployeegroupby(LOCATION)havingmax(SALARY)>=22000

SELECT\*FROMEMPloyee

selectcount(\*),LocationfromemployeewhereLocation='MP'groupby(Location),

salaryhavingsalary>=22000

selecteidfromEMPLOYEEgroupby (salary)

SELECT\*FROMEMPloyee

selectcount(\*),LocationfromemployeewhereLocation='MP'groupby(Location)

selectcount(\*),LocationfromemployeewhereLocation='MP'groupby(Location),

salaryhavingsalary>=21000

**CUSTOMIZING THE DISPLAY (User Defined Column Names)**

select 'eid '= empid, 'ename '=empname from emp;

select empid 'eid', empname 'ename ' from emp;

select empid as 'eid', empname as 'ename ' from emp;

**To Drop a Table**

drop table emp

**ORDER BY, GROUP BY, COMPUTE**

create table empl (empid int, empname char(30), empsal float, empdesi char(20));

insert into empl values (001,'sha',20000,'manager');

insert into empl values (002,'ramesh',25000,'fieldmanager');

insert into empl values (003,'karthick',20000,'asstmanager');

insert into empl values (004,'kumar',30000,'netdept');

select \* from empl;

ORDER BY

select \* from empl order by empsal asc;

select \* from empl order by empsal desc;

select \* from empl order by empid desc;

select \* from empl order by empid asc;

select \* from empl order by empname asc;

select \* from empl order by empid desc;

**TOP**

select \* from empl;

select top 2 \* from empl order by empdesi asc;

select top 2 \* from empl order by empdesi desc;

drop table empl;

**AVERAGE, COUNT, MAX, MIN, SUM, GROUPBY, COMPUTE**

drop table dept;

create table dept (proid int,proname char(20),weight int, price float, manudate varchar(10), expdate varchar(10));

insert into dept values (001,'hamam',150,'9.75','15.01.05','15.01.06');

insert into dept values (002,'lux',150,'15.75','10.02.05','10.02.06');

insert into dept values (003,'pears',150,'15.75','10.02.05','10.02.06');

insert into dept values (004,'mysore',200,'15.75','10.02.05','10.02.06');

insert into dept values (005,'santoor',150,'10.75','10.03.05','05.05.06');

insert into dept values (006,'meera',100,'15.75','10.05.05','05.03.06');

insert into dept values (007,'cinthol',100,'12.50','10.05.05','05.03.06');

select \* from dept;

**Eg : Avg**

select avg(price) from dept

select avg(distinct price) from dept

select avg(proid)from dept

**Eg : Count**

select count(\*) from dept

select count(price) from dept

select count(distinct price) from dept

**Eg : Max**

select max(price)from dept

**Eg: Min**

select min(price)from dept

**Eg : Sum**

select sum(proid)from dept

select count(\*) from dept group by weight

select count(\*) from dept where weight >75;

select count(\*) from dept where weight >100;

select count(\*) from dept where weight >75 group by weight;

select count(\*),weight from dept group by weight

select weight from dept group by weight

update dept set price =14.75 where proid=7

select weight,price from dept group by weight,price

select count(\*) from dept where weight >75 group by price;

select count(\*) from dept where weight >75 group by price having price>10;

select count(\*),price from dept where weight >75 group by price having price>10;

select count(\*) from dept where weight >75 group by price having price>13;

select count(\*) from dept where weight >75 group by price having price>16;

select count(\*) from dept where weight >150 group by price,weight having price>13;

select count(\*) from dept where weight >150 group by price,weight having price>13 and weight>160;

Example in Oracle :

string Function:

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

select upper('welcome') from dual

select lower('WELCOME') from dual

select length('Welcome') from dual

Select capitalize('welcome') from dual

select \* from dual

select substr('welcome',4,4)from dual

select current\_date from dual;

SELECT sessiontimezone, current\_date FROM DUAL;

--Next\_Day:

===========

Date of next specified date following a date

SELECT NEXT\_DAY(SYSDATE, 'FRI') FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY HH:MI:SS') FROM DUAL;

-- first day of the month

SELECT TO\_CHAR(TRUNC(SYSDATE, 'MM'), 'DD-MON-YYYY HH:MI:SS')FROM DUAL;

SELECT TO\_CHAR(TRUNC(SYSDATE, 'MON'), 'DD-MON-YYYY HH:MI:SS')

FROM DUAL;

SELECT TO\_CHAR(TRUNC(SYSDATE, 'MONTH'), 'DD-MON-YYYY HH:MI:SS')

FROM DUAL;

-- first day of the year

SELECT TO\_CHAR(TRUNC(SYSDATE, 'YYYY'), 'DD-MON-YYYY HH:MI:SS')

FROM DUAL;

select to\_char(SYSDATE,'YYYY') from dual

drop table sample

create table sample (training varchar(20) ,start\_date date )

insert into sample values ( 'Bootcamp',TO\_DATE('21.10.2013','DD-MM-YYYU'))

select \* from sample

create table dum1 (name varchar(20),price int)

insert into dum1 values(:Name ,:price)

select \* from dum1

select distinct(avg(price)) from dum1 where price=8

select avg(distinct(price)) from dum1

**Concatenating the Text Values in the Output**

drop table emp;

create table emp (empno int, empname char (30),empaddress char (50),empdesi char(20),empsal int,empphone int,);

insert emp values(001,'ram', '123abcde', 'manager', 5000, 123456);

insert emp values(002,'sam', '1234rspuram', 'asstmanager', 5000, 234567);

insert emp values(003,'sunder', '678gandhipuram', 'finance', 7500, 147852);

Select \* from emp;

select empsal+1000 from emp;

select empsal-1000 from emp;

select empsal/1000 from emp;

select empsal\*1000 from emp;

select empsal%3000 from emp;

select empsal from emp; note :(it displays the empsalary column)

select \* from emp where empno=2; note:( second row empno only )

select empsal from emp where empno=2; note:(it displays only the salary column)

select \* from emp where empsal=5500; note :(it displays the amount of value belonging 5500 )

select \* from emp where empsal=7500;

select \* from emp where empsal=4500;

select \* from emp where empaddress='1234rspuram'; note:(it displays the only the address belonging to 1234rs puram )

select \* from emp where empaddress='1234rs puram'; note :(it only displays the correct which matches )

**Having clause**

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

### SQL HAVING Syntax

### SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value GROUP BY column\_name HAVING aggregate\_function(column\_name) operator value;

The HAVING clause enables you to specify conditions that filter which group results appear in the final results.

The WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause.

## Syntax:

The following is the position of the HAVING clause in a query:

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

The HAVING clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if used. The following is the syntax of the SELECT statement, including the HAVING clause:

SELECT column1, column2

FROM table1, table2

WHERE [ conditions ]

GROUP BY column1, column2

HAVING [ conditions ]

ORDER BY column1, column2

## Example:

Consider the CUSTOMERS table having the following records:

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

Following is the example, which would display record for which similar age count would be more than or equal to 2:

SQL > SELECT \*

FROM CUSTOMERS

GROUP BY age

HAVING COUNT(age) >= 2;

This would produce the following result:

+----+--------+-----+---------+---------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+--------+-----+---------+---------+

| 2 | Khilan | 25 | Delhi | 1500.00 |

+----+--------+-----+---------+---------+

**COMPARISION OPERATORS**

Drop table employ;

Create table employ (empid int,empname char(25),empdesi char(25),empsal int,empage int);

insert employ values(001,'saradha', 'manager',25000,50);

insert employ values(002,'priya', 'asstmanager',22000,45);

insert employ values(003,'latha', 'asstmanager',22000,45);

insert employ values(004,'ramya', 'clerk',15000,48);

insert employ values(005,'bhuvana', 'ssa',5000,30);

insert employ values(006,'sita', 'offasst',2500,35);

insert employ values(007,'sudha', 'offasst',2500,32);

select \* from employ;

select empname from employ where empsal = 25000;

select \* from employ where empsal = 25000;

select \* from employ where empsal = '25000';

select empname,empsal from employ where empsal = 25000;

select empname from employ where empsal < 25000;

select empname,empsal from employ where empsal > 22000;

select empname from employ where empsal >= 22000;

select empname,empsal from employ where empsal <= 22000;

select empname,empsal from employ where empsal <> 15000;

select empname,empsal from employ where empsal != 15000;

select empname,empsal from employ where empsal !> 15000;

select empname,empsal from employ where empsal !< 15000;

select 2\*3+4 from employ where empid=1;

select 2\*(3+4) from employ where empid=1;

select empid\*empage+empsal from employ where empid=1;

select empid\*empage+empsal from employ where empid=1;

select empid\*empage+empsal from employ where empid=2;

select empid\*(empage+empsal) from employ where empid=2;

select \* from employ;

drop table employ;

**Retrieving Reconds That Match One or More Conditions**

**OR , AND, NOT**

Drop table employ;

Create table employ (empid int,empname char(25),empdesi char(25),empsal int,empage int);

insert employ values(001,'saradha', 'manager',25000,50);

insert employ values(002,'priya', 'asstmanager',22000,45);

insert employ values(003,'latha', 'asstmanager',22000,45);

insert employ values(004,'ramya', 'clerk',15000,48);

insert employ values(005,'bhuvana', 'ssa',5000,30);

insert employ values(006,'sita', 'offasst',2500,35);

insert employ values(007,'sudha', 'offasst',2500,32);

select \* from employ;

select \* from employ where empname = 'saradha' or empdesi = 'manager';

select \* from employ where empname = 'saradha' or empdesi = 'clerk';

select \* from employ where empname = 'saradha' and empdesi = 'manager';

select \* from employ where empname = 'saradha' and empdesi = 'clerk';

select \* from employ where empname = 'saradha' or not empdesi = 'manager';

**BETWEEN and NOT BETWEEN**

select \* from employ where empsal between 15000 and 25000;

select \* from employ where empsal not between 15000 and 25000;

**Adding a colum in an existing Table**

**Droping a column from an existing Table**

**LIKE, with % , \_ , [ ] , [^]**

drop table newspaper;

Create table newspaper (newspapername char(25),contactperson char(25),phone int,);

insert newspaper values('timesnewroman', 'ramesh', 123456);

insert newspaper values('Indianexpress', 'shah', 234567);

insert newspaper values('hindu', 'durai', 987654);

insert newspaper values('thinathanthi', 'palani', 654321);

insert newspaper values('t', 'sam', 012345,'fhgd');

insert newspaper values('malaimalar', 'ram', 567894,'568abcasdf');

insert newspaper values('Calaimalar', 'say', 567894,'145DEF');

select \* from newspaper;

select \* from newspaper where newspapername like 't%';

select \* from newspaper where newspapername like 'h%';

select \* from newspaper where newspapername like 't';

select \* from newspaper where phone like '1%';

select \* from newspaper where address like '%abc';

select \* from newspaper where address like '%abc%';

select \* from newspaper where contarctperson like 'sa\_';

select \* from newspaper where contarctperson like '\_am';

select \* from newspaper where contarctperson like 's[a]%';

select \* from newspaper where contarctperson like 's[h]%';

SELECT \* FROM Customer WHERE FirstName like '[a-m]%r'

select \* from newspaper;

drop table newspaper;

Note:(we have to change a value in a particular cell )

update newspaper set newspapername= 'timesofindia' where phone = 123456;

update newspaper set newspapername= 'indianexpress' where phone = 234567;

Note:(we have to remove a particular row)

delete from newspaper where phone = 123456;

Note:(we have to add a column in table)

alter table newspaper add address char(40);

Note:(to drop a particular column)

alter table newspaper drop column address;

Note:(to add a value for that particular cell)

update newspaper set address= '123abc' where phone = 123456;

update newspaper set address= '321abc' where phone = 234567;

update newspaper set address= '456abc' where phone = 987654;

update newspaper set address= '654abc' where phone = 654321;

Note:(to add a particular column column name)

alter table newspaper add newsid int;

Note:(to drop a particular column)

alter table newspaper drop column newsid ;

**Retrieving Records That Contain NULL Values**

drop table test;

Create table test(eid int,eadd varchar(25),phone int,);

insert test values('001', '123abc', 123456);

insert test values('002', '234bcd', 234567);

insert test values('003', '345cde', 987654);

insert test values('004', '456def', 654321);

select \* from test;

alter table test add ename char(25);

insert test values('005', '567efg', 012345,'ram');

select \* from test;

select \* from test where ename is null;

select \* from test where ename is not null;

**STRING FUNCTIONS**

select ASCII('ABC') Note :(it returns a value of the 1st left most

character A)

select ASCII('BCA')

select ASCII('s') Note :(it returns a value character s)

select ASCII('S')

select CHAR(67)

select ASCII('/')

select ASCII('\')

select ASCII(' ')

select ASCII('.')

select CHARINDEX('E','HELLO')

select CHARINDEX('L','HELLO')

select CHARINDEX('EL','HELLO')

select CHARINDEX('A','HELLO')

select DIFFERENCE('HELLO','HELL')

select DIFFERENCE('HELLO','HE')

select DIFFERENCE('HELLO','HEL')

select DIFFERENCE('H','HELL')

select DIFFERENCE('HELLO','L')

select DIFFERENCE('HELLO','H')

select DIFFERENCE('HELLO','A')

select DIFFERENCE('HELLO','')

select DIFFERENCE('sha','sha')

select left('shafeek',3)

select left('shafeek',6)

select left('shafeek',8)

select len('kumarappan')

select len('radha')

select lower('RAMESH')

select lower('Ramesh')

select upper('ramesh')

select ltrim (' balaji')

select ltrim (' balaji ')

select rtrim (' balaji ')

select ltrim (' ajay')

select patindex ('%box','actionbox')

select patindex ('%box','messagebox')

select patindex ('%ebox','messagebox')

select patindex ('%box%','postboxoffice')

select patindex ('box%','postboxoffice')

select patindex ('box%','boxoffice')

select patindex ('%kumar','krishnakumar')

select reverse ('rahamathulla')

select reverse ('babu')

select right ('suganya',4)

select 'Krishna'+space(2)+'Kumar'

select 'KrishnaKumar'

select 'Krishna Kumar'

select str (123.45,6,2)

select str (123.45,4,2)

select str (23.45,4,2)

select str (123.45,3,2)

select str (123.4)

select str (124.57)

select stuff ('weather',2,2,'i')

select stuff ('weather',1,1,'l')

select stuff ('shafeek',1,2,'R')

selectstuff ('weather',2,1,'ii')

select substring ('weather',2,2) Note :(it displays the second 2 char’s)

select substring ('weather',2,3) Note :(it displays the second 3 char’s)

**DATE FUNCTIONS**

select getdate(); Note :(it displays the current system date , time )

select datediff(mm,'01.01.10',getdate()); Note : MM/DD/YY Format

select datediff(mm,'05.01.10',getdate());

select dateadd(dd,5,getdate());

select dateadd(dd,10,getdate());

select dateadd(mm,5,getdate());

select dateadd(mm,1,getdate());

select dateadd(yy,5,getdate());

select dateadd(yy,1,getdate());

select datename(mm,getdate());

select datepart(dd,getdate());

select datepart(mm,getdate());

select datepart(yy,getdate());

select datepart(yyyy,getdate()); Note :(it displays the current year)

select datepart(qq,getdate()); Note :(it displays the current quarter)

select datepart(mm,getdate()); Note :(it displays the current month)

select datepart(dy,getdate()); Note :(it displays the current day (1-365))

select datepart(wk,getdate()); Note :(it displays the current week of this year(1-51))

select datepart(ww,getdate()); “ “ “ “

select datepart(dw,getdate()); Note :(it displays the current weekday)

select datepart(hh,getdate()); Note :(it displays the current hour)

select datepart(mi,getdate()); Note :(it displays the current minutes)

select datepart(n,getdate()); “ “ “ “

select datepart(ss,getdate()); Note :(it displays the current seconds)

select datepart(s,getdate()); “ “ “ “

select datepart(ms,getdate()); Note :(it displays the current milli sec)

**MATHEMATICAL FUNCTIONS**

select abs (1234.5) note :(it displays the absolute value)

select abs (1234.3)

select sin(30) note :(it returns sin of the angle in radians )

select asin(1) note:(it only assign –1 to +1,

it returns the angle in radians)

select cos(30) note :( it returns sin of the angle in radians)

select Acos(30)

select sin(45) note :( it returns sin of the angle in radians )

select asin(45)

select tan(30) note :(it displays the tan30 value)

select atan(30)

select cot(30) note :(it displays the cot30 value)

select acot(30)

select degrees(10) note:(it returns the smallest integer greater than or equal to the specified value)

select degrees(20)

select exp(45.05) note :(returns the exponential value of specified value)

select exp(10.10)

select exp(0.01)

select exp(1.25E-4)

select floor(45.05) note:(it returns the largest integer less than or equal to the specified value)

select floor(47.50)

select ceiling(45.06) note:(it gives the upper value to the given value)

select ceiling(47.50)

select log(5) note:(natural logarithm)

select log(5.05)

select log(12.22)

select log10(12.22)

select log10(5)

select log10(5.05)

select pi() note:(it returns the constant value)

select power(10,3) note:( it returns the value of numeric expression)

select power(10,4)

select power(20,3)

select radians(90) note:(it converts degrees into radians)

selectradians(180)

select rand(45) note:(it returns a random float number between 0 and 1 )

select rand(60)

select rand()

select round (1234.567,2) note:(it rounds the value after the decimal )

select round (1234.567,1)

select round (1234.567,0)

select round (1234.567,-1)

select round (1234.567,-2)

select round (1234.567,-3)

select sign (0) note:(it returns positive, negative and zero)

select sign (20)

select sign(100)

select sign (-20)

select sign(-18)

select sqrt (4) note:(it returns the square root of the given value)

select sqrt (16)

select sqrt (25)

select sqrt (100)

**User System Functions**

select host\_id() Note:(it returns the current host process id of

client process)

select host\_name() Note:(it returns the current host computer name of

client )

select suser\_sid() Note:(it returns the security identification

number(SID)corresponding to the user’s login name )

select suser\_sid('user1') Note:(it returns the user’s login name )

select suser\_sname(0xC237EC3462C7F141BDBB0132B654D429)

Note:(it returns the user’s login name

corresponding to the username )

select user\_id('dbo') Note:( it returns the username corresponding to

the database identification number)

select user\_name(0xC237EC3462C7F141BDBB0132B654D429)

Note:(it returns the database id number of the

address)

select user\_name(1)

select db\_id('sha') Note:(it returns the database id)

select db\_name(1) Note:(it returns the database name)

select object\_id ('sysobjects')

Note:( it returns the database object id number)

select object\_name(1) Note:(it returns the database object name )