

Experiment NO. :- 02

Aim: To study DDL commands on SQL.

common DDL commands:

command

Description

CREATE

Create database or its objects
(table, index, function, views,
STORE, procedure and triggers)

DROP

Delete objects from the
database

ALTER

Alter the structure of the
database

TRUNCATE

Remove all records from a table,
including all spaces allocated for
the records are removed.

COMMENT

Add comments to the data
dictionary

RENAME

Rename an object existing
in the database.

SQL> select * from abc;
SP2-0640: Not connected
SQL> connect

Enter username: system
Enter password: cap
connected

(1) Create Table

Create Table student
(RN number (2),
Name Varchar2 (10),
Email Varchar2 (15);
(constraint RN primary key)

O/P:- Table created.

SQL> DESC student

RN	Name	Email
10	student	student@gmail.com

(2) ALTER

After Table student
Add (rn number(10));

Desc student;

O/P:- RN Name Email MN;

2.2) ALTER Table student

Modify (Email(Varchar(20));
Table altered;

O/P:- Table altered;

Desc student

Q1:-

RN	NAME	Email	MN
90101201	90101201	90101201	90101201
		90101201	90101201

Q.3) DROP column MN of student table.

ALTER table student drop column MN;

Drop (Email);

Q1:- table altered.

RN	NAME	EMAIL
90101201	90101201	90101201
		90101201

2.4) Rename

ALTER Table student

Rename column rn to stn.

O/P:- Table altered

DESC student

O/P:-

Srn	Name	mn
1	John	9090
2	Adam	9090

(3) Rename

rename student to students

DESC students

O/P:-

Srn	Name	mn
1	John	9090
2	Adam	9090

4) Truncate :-

sql> Truncate table students;

o/p:- table truncated

sql> Select * from students;

o/p:- no rows selected.

5) Drop :-

sql> Drop table students;

o/p:- table dropped

sql> Desc students;

o/p: ORA-00942 : table or view does not

+ Conclusion:

Experiment No. 03

Aim: To study Data manipulation language (DML) command in SQL.

SQL> SELECT * FROM abc;
SP2-0640 : Not connected
SQL> DESC abc
SP2-0641b :- Not connected

SQL> CONNECT
Enter user-name : system
Enter password : dbms
connected.

SQL> CREATE TABLE student
(name VARCHAR(20), roll-no NUMBER(2),
id NUMBER(6), branch VARCHAR(10));
Table created

SQL> DESC student

NAME	NULL	TYPE
NAME		VARCHAR
ROLL-NO		NUMBER
ID		NUMBER
BRANCH		VARCHAR

SQL> INSERT INTO student VALUES
21 'Sam', 72, 308329, 'IT')
row created.

SQL > select * from student;

NAME	ROLL-NO	ID	BRANCH
Sam	72	308029	IT

SQL > insert into student (name, branch) values

2 ('Rahul', 'IT')

1 row created

NAME	ROLL-NO	ID	BRANCH
Sam	72	308029	IT
Rahul			IT

Scripted

SQL > insert into student values

2 ('&name', &rollno, &id, '&branch');

Enter values of name : Sid

Enter values for roll-no: 69

Enter values id : 309088

Enter values for branch: IT

01d 2 :- ('&name', &roll-no, &id, '&branch')

01d 2, - ('Siddhant', 69, 309088, IT)

1 row created

NAME	ROLL-NO	ID	BRANCH
Sam	72	308029	IT
Rohan	65	308029	IT
Sid	69	309088	IT

* Possibly scripted

SQL> INSERT INTO student (roll-no, id) VALUES (74, 690844)

Enter value for roll-no : 74

Enter value for id : 690844

1 row added

SQL> SELECT * FROM student

Name	ROLL-NO	ID	BRANCH
Sam	65	308329	IT
Rohan	69	309081	IT
Sid	74	690844	IT

4 rows selected.

SQL> Insert into student values

= ('Vaibhav', 86, 609797, 'EXT');

SQL> Insert into student values

= ('Sam', 65, 309080, 'IT');

1 row created.

SQL> Insert into student

= (309876, 'Mechanical');

1 row created.

SQL > Select * from student;

Name	ROLL-NO	ID	BRANCH
Sam	65	309080	IT
Rohan	69	309081	IT
Sid	74	690844	IT
Vaibhav	56	609797	EECE
Sam	65	309080	IT
Mayur	43	687969	Electrical
		309870	mechanical

8 rows selected

SQL > Update student

2 SET NAME = 'Sagar'

3 WHERE BRANCH = 'Mechanical';

* All rows' update

SQL > Select * from student

Name	ROLL-NO	ID	BRANCH
Sam	65	309080	Civil
Rohan	69	309081	IT
Sid	74	690844	IT
Vaibhav	56	609797	EECE
Sam	65	309080	EECE
Mayur	43	687969	Electrical
Sagar			mechanical

8 rows update.

SQL > SELECT * from student;

* few new update

NAME

NAME	ROLL-NUMBER	ID	BRANCH
Shm			
Rohan	65	309080	IT
Sid		309081	IT
Vishnu	69	309081	IT
Sam	74	690844	IT
Mayur	56	309080	IT
Sagar	65	687969	IT
	48	309876	IT

SQL> Delete from student

2 where roll-no = 74;

1 row deleted

* One row deletion

SQL> Select * from student

NAME	ROLL-NUMBER	ID	BRANCH
Sam	65	309080	IT
Rohan		309081	IT
Vishnu	69	690844	IT
Sam	56	309080	IT
Mayur	65	687969	IT
Sagar	48	309876	IT

5 rows selected.

* few rows deletion

SQL> DELETE from student

where name = 'Sam'

2 rows deleted.

SQL> DELETE * from student;

5 rows deleted.

SQL> Select * from student;

no rows selected.

SQL> create table info

(Name varchar(10), Id number(10), salary number)

Table created.

SQL > Create Table info2

(Name varchar(10), id number(10), salary number)

Table created!

SQL > Insert into info2 values ('Sam', 65, 45000);

1 row created.

SQL > Insert into info2 values ('Rohan', 61, 55000);

1 row created.

SQL > Select * from info2;

Name	id	Salary
Sam	65	45000
Rohan	61	55000

SQL > Insert into info2 values ('siddhant', 69, 65000);
1 row created.

SQL > Insert into info2 values ('Sam', 60, 75000);
1 row created.

SQL > Select * from info2;

Name	id	Salary
Siddhant	69	65000
Sam	60	75000

Merge function

SQL > Merge into info2 a

2 using info2 b

3 on (a.name = b.name)

4 when matched then

5 update set a.id = b.id, a.salary = b.salary

6 when not matched then

7 insert values (b.name, b.id, b.salary)

2 rows merged.

SQL > select * from info2;

Name	ID	Salary
Siddhant	69	65000
Sam	65	45000
Rohan	61	55000

* Conclusion:

- we successfully implemented the DMLs to insert rows in table, delete rows in a table, merge rows in a table,

Experiment no: 4

Aim: To write basic SQL select statements.

SQL> connect

Enter username: system
Enter password: dbms

connected

SQL> create table student (Roll-no number(2),
Name varchar(20), Age Number(2),
marks number(3));

Table created.

SQL> desc student

Name	Null?	Type
Roll-no	Null?	Number(2)
Name		Varchar(20)
Age		Number(2)
Marks		Number(3)

SQL> Insert into student values (1, 'Rom', 20, 85);
1 row created.

SQL> Insert into student values (2, 'siddhant', 21, 75);

SQL> ~~Insert into student values (3, 'Vedant', Null, 75);~~
1 row created

SQL> ~~Insert into student values (4, 'Atharva', 20, 80);~~
1 row created

SQL> ~~Insert into student values (5, 'Ashishi', 21, 78);~~
1 row created

* Selecting all columns

SQL> select * from student;

roll-no	name	age	marks
1	ram	20	85
2	siddhant	21	90
3	Vedant	20	75
4	Atharva	20	80
5	Ashish	21	90

* Selecting specific columns

SQL> select name, marks from student;

Name	marks
ram	85
Siddhant	90
Vedant	75
Atharva	80
Ashish	90

* Arithmetic operations (+, -, *, /)

SQL> select name, marks + 10 * 2, age - 2 / 2, from student;

Name	marks + 10 * 2	age - 2 / 2
ram	105	19
Siddhant	110	20
Vedant	95	
Atharva		19
Ashish	110	20

* Operator precedence

SQL> Select name, marks + 10 - 5 * 2 / 2 from students;

Name	marks + 10 - 5 * 2 / 2
ram	90
Siddhant	95
Vedant	80
Adharva	90
Ashish	95

* Using parenthesis (to change precedence)

SQL> Select roll-no, (marks + 10) * 2 / 2 from students;

Roll-no	(Marks + 10) * 2 / 2
1	190
2	200
3	170
4	190
5	200

* Alias (remaining column for output)

SQL> Select name as student-name, marks as scored-marks from students;

student_name	scored_marks
RAM	85
Siddhant	90
Vedant	75
Atharva	90
Ashish	90

② Concatenating two columns.

SQL select roll_no || name from student;

roll_no name
1 RAM
2 Siddhant
3 Vedant
4 Atharva
5 Ashish

③ Concatenating two columns with string in between

SQL select name " is having marks " || marks from student;

Name		is having marks	marks
RAM	is having marks		
Siddhant	is having marks		85
Vedant	is having marks		90
Atharva	is having marks		75
Ashish	is having marks		90

* SQL allows duplicate values

SQL> Select marks from student;

MARKS
85
90
75
90

* eliminating duplicate values

SQL> select distinct name, marks from student;

Name	Marks
ram	85
Siddhant	90
Vedant	75
Atharva	

Conclusion:- In this practical we learned about specific columns, performing arithmetic operations with operator, precedence, with alias, combination of columns and eliminating duplicates in SQL.

Aim: to insert data from multiple tables.

SQL> connect

Enter user_name: system

Enter password: dbms

Connected.

SQL> Create table emp (Eid number(2),
Name varchar(3), DID Number(3),
JID Number(3), Salary number(5));
Table created.

SQL> desc emp

Name	Null	Type
Eid	Null	Number(2)
Name	Null	Varchar(3)
DID	Null	Number(3)
JID	Null	Number(3)
Salary	Null	Number(5)

SQL> insert into emp values (1, 'A1', 100, 501, 1000);
1 row created.

SQL> select into emp values (2, 'B', 101, 502, 20000)

SQL> select * from emp;

EMPID	NAME	DID	LID	Salary
1	A	100	501	10,000
2	B	101	502	20,000

SQL> create table Dept (DID Number(3), Dname
 varchar(7), LID Number(3));

Table created.

SQL> insert into dept values (100, 'HR', 50);

1 row created.

SQL> insert into dept values (102, 'Admin', 51);

1 row created.

SQL> select * from dept;

DID	DNAME	LID
100	HR	50
102	ADMIN	51

SQL> create table loc (LID Number(3), Lname
 varchar(8));

Table created.

SQL> desc loc;

NAME	TYPE
LID	Number(3)
LNAME	Number(3)

SQL> create table loc (ID number(3), LName, Valence(5));

Table created.

SQL> des loc;

NAME	NULL	TYPE
LID		NUMBER(3)
LNAME		NUMBER(5)

SQL> insert into loc values (51, 'XYZ');

1 row created

SQL> insert into loc values (52, 'PQR');

1 row created.

SQL> select * from loc;

LID	LNAME
51	XYZ
52	PQR

④ SQL> select * from emp,dept;

EID	NAME	DID	JID	Salary	DID	NAME
1	A	100	501	10,000	100	HR
1	A	100	501	10,000	102	Admin
2	B	101	502	20,000	100	HR
2	B	101	502	20,000	102	Admin

[Cartesian product]

SQL > select emp.DID from emp,dept;

DID

100

100

101

101

[equi join]

④ Select * from emp,dept;
where emp.DID = dept.DID;

EID NAME DID JID SALARY DID DNAME C1
1 A 100 S01 10000 100 HR S0

SQL > Select * from emp,dept,loc;

EID	NAME	DID	JID	SALARY	DID	DNAME	LOC
1	A	100	S01	10000	100	HR	S0
1	A	100	S01	10000	100	HR	S0
1	A	100	S01	10000	102	Admn	S1
1	A	100	S01	10000	102	Admn	S1
2	B	101	S02	20000	100	HR	R0
2	B	101	S02	20000	100	HR	R0
2	B	101	S02	20000	102	Admn	S1
2	B	101	S02	20000	102	Admn	S1

④ SQL > Select * from emp,dept,loc
where emp.DID = dept.DID
and dept.DID = loc.DID;
[Joining more than
2 tables]
no rows selected

SQL> delete from loc

where lid = 51 ;

1 row deleted.

SQL> insert into loc values (50, 'XYZ')

1 row created

SQL> select * from emp,dept,loc

where emp.did = dept.did

and

dept.lid = loc.lid ;

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50

SQL> select * from emp,dept [Right outer join]
where emp.did (+) = dept.did ;

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50
-	-	1	-	-	102	Admin	51

SQL> select * from emp,dept [Left outer join]
where emp.did = dept.did (+) ;

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50
2	B	101	502	20,000	-	-	-

(Self Join)

④ SQL> select * from emp e JOIN emp e USING (e1)

EID	NAME	DID	JID	SALARY	NAME	DID	JID
1	A	100	501	10,000	A	100	501
2	B	101	502	20,000	B	101	502

[Cross Join]

④ SQL> select * from emp CROSS JOIN dept;

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50
1	A	100	501	10,000	102	Admin	51
2	B	101	502	20,000	100	HR	50
2	B	101	502	20,000	102	Admin	51

(Natural Join)

④ SQL> select * from dept,10c

where dept.lid=10c.lid;

DID	DNAME	LID	LID	NAME
100	HR	50	50	XYZ

SQL> select eid from emp,dept

where emp.did=dept.did;

↓

[Using column name]

④ SQL> select e.eid,e.name,e.did,d.did,d.lid
from emp e join dept d
on (e.did=d.did);

EID	NAME	DID	DID	LID
1	A	100	100	50

[Left outer join]

④ SQL > Select * from emp e Right outer join dept d
on (e.did = d.did)

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50
-	-	-	-	-	102	Admin	51

⑤ SQL > Select * from emp e full outer join dept d
on (e.did = d.did)

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
1	A	100	501	10,000	100	HR	50
2	B	101	502	20,000	102	Admin	51

⑥ SQL > Select DID from emp, dept
where emp.did between dept.did .min - did
and dept.max - did; (Non-Equi²Join)

EID	NAME	DID	JID	SALARY	DID	DNAME	LID
2	B	101	502	20,000	102	Admin	51

SQL > Disconnect

Disconnected,

Conclusion: Display data from multiple tables is a fundamental concept of SQL. Using joins such as, Cartesian product, Equi join, Non Equi join, Outer joins Left, Right, full

Experiment no. 06

Group function,

Aim: To aggregate data using

Create table employee

emp_id	int(3)
emp_name	varchar(10)
dept_id	int(5)
salary	int(10)
exp	int(2)
job-title	varchar(20)

);

table created.

Insert into employee values (1, 'Alice', 101, 50000, 3,

Insert into employee values(2, 'Ram', 102, 60000, 5,)

Insert into employee values(3, 'Raj', 103, NULL, 6,)

(1) sum

select sum (salary) from employee;

sum (salary)

110000

(2) Average = (Avg(1))

Select Avg(salary) from employee;

Avg (salary)

55000.00

(3) Min()

Select MAX(salary) from employee;

MAX(salary)

60000

(4) MAX()

Select MAX(salary) from employee;

MAX(salary)

60000

(5) COUNT() - single count

Select (dept_id) from employee

Count (dept_id) from employee

Count (dept_id)

3

(6) COUNT(*) → counting all rows

Select COUNT(*) from employee;

Count(*)

3

(7) COUNT() → using distinct keyword

Select COUNT(DISTINCT dept_id), from employee;

Count(DISTINCT dept_id);

3

(8) NVL() → number

Select emp_name, NVL(salary, 60000) as final salary
from employee;

emp_name

Alife

Ram

Roj

final_Salary

50000

60000

70000