

Assignment - 1

1. Write a query to create a table employee with empno, emp ename, designation and salary

SQL > create table employee (empno number (10), ename varchar2(10), designation varchar2(10), salary number (8,2));

Table created

2. Write a query to display the column name and data type of the table employee.

SQL > desc employee;

Name	Null ?	Type
EMPNO		NUMBER(10)
ENAME		VARCHAR2(10)
DESIGNATION		VARCHAR2(10)
SALARY		NUMBER(8,2)

3. Write a query to create a table from an existing table with all the fields.

SQL > create table employee-cop as select * from employee;

Table created

4. Write a query to create table from an existing table with selected fields.

SQL > create table employee-copy as select empno, salary from employee;

Table created

5) Write a query to create a new table from an existing table without any record.
SQL> create table employee_copy1 as select * from employee where 1=2;

Table created.

6) Write a query to alter the column empno number(4) to empno number(6).

SQL> alter table employee modify empno number(6);

Table altered.

7) Write a query to alter the table employee with multiple columns (empno, ename).

SQL> alter table employee modify (empno number(10), ename varchar2(15));

Table altered.

8) Write a query to add a new column in employee table.

SQL> alter table employee add DOJ date ;

Table altered.

9) Write a query to add multiple columns in employee table.

SQL> alter table employee add (DOB date, age number(2));

Table altered.

10) Write a query to drop a column from an existing table employee.

SQL > alter table x.emp10 drop column empno;

Table altered.

11) Write a query to drop multiple columns from the employee table.

SQL > alter table x.emp10 drop (ename, age);

Table altered.

12) Write a query to rename table employee to emp.

SQL > RENAME employee To emp-loyee;

Table renamed.

Final Table after all operations -

SQL > desc emp-loyee;

Name	Null?	Type
DESIGNATION		VARCHAR(10)
SALARY		NUMBER(8,2)
DOJ		DATE
DOB		DATE

Assignment -2

- ① SQL > `create table employee (emp_id number(10), f_name varchar(10), l_name varchar(10), job_type varchar(10), salary number(10), dept varchar(10), commission number(10), manager_id & number(10));`
- ② SQL > `desc employee;`
- ③ SQL > `alter table employee add primary key (emp_id);`
SQL > `desc employee;`
SQL > `alter table employee modify (f-name not null, salary not null);`
SQL > `desc employee;`
- ④ SQL > `alter table employee add date-of-joining date;`
SQL > `desc employee;`
- ⑤ SQL > `create table location (loc_id number(10), city varchar(10), contact_no number(10));`
SQL > `desc location;`

- ⑥ SQL > alter table location modify city varchar(5);
- ⑦ SQL > desc location;
- ⑧ SQL > alter table location drop column contact_no;
- ⑨ SQL > desc location;
- ⑩ SQL > alter table department rename column d-name to dept;
- ⑪ SQL > alter table location rename column city to address;
- ⑫ SQL > desc location;
- ⑬ SQL > rename location to loc;
- ⑭ SQL > desc loc;
- ⑮ SQL > alter table loc modify address varchar(10);
- ⑯ SQL > /
- ⑰ SQL > truncate table loc;
- ⑱ SQL > drop table loc;
- ⑲ SQL > select * from department;
- ⑳ SQL > select f-name, l-name, salary, salary + 1000 from employee;

②1 SQL > Select salary + 1000 , salary + 12 * 100 from employee ;

②2 SQL > Select f-name as name and salary as annual from employee ;

Select f-name as name and salary as annual from employee ;

②3 SQL > Select f-name as name , salary as annual from employee ;

②4 SQL > Select emp-id , f-name , l-name , job-type from employee where salary = (select max (salary) from employee) ;

②5 SQL > select emp-id , f-name , l-name , job-type from employee where salary = (select min (salary) from employee) ;

②6 SQL > Select avg (salary) from employee ;

Assignment-3

- 1) SQL > Select f-name, l-name , job-type from employee ;
- 2) SQL > Select f-name , ' ' , l-name , ' is a ' ,
dept "Employee details " from employee ;
- 3) SQL > Select f-name || 'monthly' , 'salary "Employee
details " from employee ;
- 4) SQL > Select d-name from department ;
- 5) SQL > Select f-name , l-name from employee where
dept = 'sales' ;
- 6) SQL > Select f-name , l-name from employee where
salary > 50000 ;
- 7) SQL > Select f-name , l-name from employee where
manager_id > 1 ;
- 8) SQL > Select f-name , l-name from employee where
salary > 40000 or salary < 70000 ;
- 9) SQL > Select f-name , l-name from employee
where manager_id > 0 and manager_id < 9 ;
- 10) SQL > Select f-name from employee where l-name
Like 'K%' ;

11) SQL > Select f-name, salary from employee where
② l-name like 'K%' and l-name like '%R';

12) SQL > select f-name from employee where l-name
(like '---O%');

13) SQL > select f-name, l-name from employee where
job-type = 'engineer' and salary > 50000;

14) SQL > select f-name, l-name from employee where
dept = 'production' and salary > 60000;

15) SQL > select min(salary) from employee;
SQL > select min(salary) from employee;

SQL > Select sum(salary) from employee;

SQL > Select avg(salary) from employee;

SQL > Select avg(salary) from employee;

16) SQL > select l-name, f-name from employee;

17) SQL > select dept, count(dept) from employee
group by dept;

18) SQL > select count(dept) from employee;

19) SQL > select avg(commission) from employee;

Assignment - 4

- i) SQL > Select upper ('Joy mohunbagan').
- SQL > Select lower ('Joy mohunbagan')
- SQL > Select concat (fname, lname) from stu-data;
- SQL > Select instr ("sch.com", "g") as matchpartition;
- SQL > Select len ('mohunbagan');
- SQL > Select sqrt (opening-amt) from customer;
- g) a) SQL > Select power (A,2);
- b) SQL > Select ceil (25.75);
- c) SQL > Select substr ("Joy mohunbagan", 5,3)
- d) SQL > Select extractstring;
- e) SQL > Select max (price) as largestprice from products;
- f) SQL > Select min (price) as smallestprice from products;
- g) SQL > Select round (235.2,2) as roundvalue;
- h) SQL > Select avg (price) from products
- i) SQL > Select exp (1);

- 11) 4) (a) SQL > select ceiling (14.887) as ~~ceilval~~ ;
SQL > select floor (14.887) as floorvalue ;
(b) SQL > select power (8,7);
- 8) SQL > select getdate();
- 6) set SQL > select * from employee where extract
(year from age) (current_date, hire_date) = 1;
- 7) (a) Select to_char (add_months (hire_date, 1),
'DD-MON-YYYY') "Next Month" from dual;
(b) Select next_day ('12-May-2015', 'MONDAY')
"Next day" 2 from dual;
- (c) SQL > select last_day ("2022-09-09")
- (d) SQL > select round (1.1, 2);
- (e) SQL > select truncate (135.233, 2);
- 8) (a) SQL > update employee set salary = salary +
(salary * 10/100);
(b) SQL > update employee set marketing =
marketing + (marketing * 20/100);

Assignment-5

- 1) SQL > select * from employee cross join department;
- 2) SQL > select Ename, Lname from employee E cross join department d where e.dep-id = d.dep-id;
- 3) SQL > select Ename, date-of-joining from employee E cross join department d where e.dep-id = d.dep-id = 'Delhi';
- 4) SQL > create table Emp_address (Emp-id number(10), city varchar(20), district varchar(20), state varchar(20));
- 5) SQL > select Ename, from employee E join department D on E.department-id = D.department-id join location L on D.Emp-Address-id = L.Emp-Address-id;
- 6) SQL > select D.Name from department;
- 7) SQL > create table Job-Grades (Grade varchar(5), lowest-sal number(20), highest-sal number(20));
- 8) SQL > select Ename, Lname, E.salary, J.grade-level from employee E join Job-Grades J on E.salary between J.lowest-sal and J.highest-sal;
- 9) insert into employee (13, 'Venugopal', 'Iyer', 'Salesman', 1, 1, 'Marketing2', 3, 12-Jan-2014);

- 10) insert into department ('Marketing 2', 'ICAI', 5);
- 11) a) SQL> Select * from employee natural join department;
b) SQL> Select * from employee E inner join
department D on D.did = E.did;
- c) SQL> Select * from employee E inner join
department D on D.did = E.did;
- d) SQL> Select * from employee E right join
department D on D.did = E.did;
- e) SQL> Select * from employee E full outer
join department D on D.did = E.did;

Assignment-6

real', 5)

department

- 1) SQL> Select first_name as upper (worker_firstname)
from marker;
- 2) SQL> Select substr(first-name, 1, 3) from worker;
- 3) SQL> Select position("A" in first-name) from worker
where first-name = "Amitabh";
- 4) SQL> Select ~~concat~~ (first-name, " ", last-name) as
complete-name from number;
- 5) SQL> Select * from worker order by first-name
ASC, department DESC;
- 6) SQL> Select * from worker where first-name
like "%a%";
- 7) SQL> Select * from worker where first-name
like "%a";
- 8) SQL> Select * from worker where salary between
100000 AND 600000;
- 9) SQL> Select count(*) from worker where
department = "Admin";
- 10) SQL> Select department, count(worker-id) No.
of-workers from worker group by department,
order by no.-of-workers DESC;

- 11) SQL > Select distinct w.
from worker w inner join title
= t. worker-ref-id AND t. worker-title in ('Manager')
MOD(worker-id, 2)
- 12) SQL > Select * from worker where
< 0;
- 13) SQL > Select * from worker MINUS select * from
title;
- 14) SQL > Select * from worker order by salary DESC
limit 10;
- 15) SQL > Select distinct w. worker-id, w. first-name
w. salary from worker w, worker w1 where #
w. salary = w1. salary and w. worker-id != w1. work
id
- 16) SQL > Select department, count(department) as 'num
of workers' from worker group by department;
- 17) SQL > Select t.department, t.first-name, t.salary
from (select man (salary) as totalsalary, departm
from worker group by department) as tempnew
inner join worker t on tempnew.department =
t.department and tempnew.totalsalary = t.salary';
- 18) SQL > Select department, sum (salary) from
worker group by department;

Assignment - 7

- 1) SQL > create table Job_History (emp_id number(2), start_date date, end_date date, job_type varchar(20), D_name varchar(20));
- 2) SQL > select job_types from employees;
- 3) SQL > select department, job_type from employees;
- 4) SQL > select emp_id, job_type from employees where previous_i = current_id;
- 5) SQL > select emp_name from employees as employee
join salary as salary on salary.empno. = employee.
MGR where employee.sal > chitraSal';
- 6) SQL > select emp_id, ename from employee where
change = "new job";
- 7) SQL > Select emp_id, ename from employee where
emp_id = + join job_History as HTT . where job-type
= "same";
- 8) SQL > select emp_id, ename from employee where
salary < 10000 and salary < 70000 ;
- 9) SQL > select lname, job_type, salary from
employees where salary = 35000;

110) SQL
select job-type from employees where
salary = 10000;

11) SQL
select department from employees where
salary > sales;

, 12) SQL
select * from employees where salary =
(select min(salary) from employees);

13) SQL
Select * from employees where salary >
all (select avg(salary) from employees group
by department_id = "Engineer");

Assignment - 8

Q (i) SQL > create table author 1 (author 1 - id int,
author 1 - name varchar (20), author 1 - city varchar(20),
primary key (author 1 - id));

SQL > create table publisher 1 (publisher 1 - id int,
publisher 1 - name varchar (20), publisher 1 - city
varchar (20), publisher 1 - country varchar (20),
primary key (publisher 1 - id));

SQL > create table category 1 (category 1 - id int,
description varchar (30), primary key (category 1 - id));

(ii)
SQL > create table catalogue 1 (book - id int - primary
key ; book - title varchar (30), author 1 - id int,
publisher 1 - id int, category - id int, year int,
price int, foreign key (author 1 - id) references author 1
(author 1 - id), foreign by (publisher 1 - id) references
publisher 1 (publisher 1 - id), foreign key (category - id)
references category 1 (category - id));

SQL > create table order details 1 (order - id int,
book id int, quantity int, primary key (order - id),
foreign key (book - id) references catalogue 1 (book - id));

(ii) SQL > select * from author1 where author1-id
in (select author1-id from catalogue1 where
year > 200 AND price > (select avg(price) from
catalogue1) group by author1-id having count(*) > 1);

(iv) SQL > Select author1-name from author1 a. catalogue1
e where a.author1-id = e. author1-id AND
book-id in (select book-id from order details1
where quantity = select max(quantity) from order
details1);

(v) SQL > update catalogue1 set price = 1.1 * price where
publisher1-id in (select publisher1-id from publisher1
where publisher1-name = 'pearson');

Ques (iii) SQL > Select customer-name from depositon
where acc-no in (select acc-no from account
branch-name = 'mumbai') group by acc-no having
account(~~acc-no~~) > 2;

(iv) SQL > Select customer-name from customer
where acc-no in (select acc-no from account
where branch-name in (select branch-name from
Branch where Branch-city like '%Bangalore'));

(N) SQL γ delete acc-no from account where
branch-name in / select branch-name, branch-city
from branch where branch-city like 'Bangalore'
group by branch-city);

3)(iii) SQL γ select c. custno., count(*) as no.-of-orders,
avg(o. ord-amt) as avg-order-amount from customer
c, order 1 o where o.customer = c.custno group by
c.custno.;

(iv) SQL γ select s.orderno from shipment s, warehouse w
where s.warehouse = w.warehouseno and w.city =
'Bangalore' group by orderno having
count(*) and not count(*) = 0;

(v) SQL γ delete from item where itemno = 3;

Assignment - 9

4) Write a PL/SQL program to check whether given number is prime or not.

Set server output on

```
| declare  
|   n number;  
|   i number;  
|   flag number;  
| begin  
|   i := 2;  
|   flag := 1;  
|   n := &n;  
|   for i in 2....n/2  
|   loop  
|     if mod(n, i) = 0  
|     then  
|       flag := 0;  
|       exit;  
|     end if;  
|   end loop;  
|   if flag = 1  
|   then  
|     dbms_output.put_line('prime');  
|   else  
|     dbms_output.put_line('not prime');  
|   end if;  
| end;
```

* write a plsql program to check the given year is leap year or not

5) Write a PL/SQL program to generate fibonacci series upto N.

→ Set serveroutput on

declare

first number := 0;

Second number := 01;

third number;

n number := &n;

i number;

begin

dbms_output.put_line('fibonacci series is :');

dbms_output.put_line(first);

dbms_output.put_line(second);

for i in 2...n

loop

third := first + second;

first := second;

second := third;

dbms_output.put_line(third);

end loop;

end;

/

6) Write a PL/SQL program for calculating sum of two numbers.

Set serveroutput on

declare

var1 integer;

var2 integer;

var3 integer

begin

var1 := & var1;

var2 := & var2;

var3 := var1 + var2

dbms_output.put_line(var3);

end;

/

Q7) Write a PL/SQL program to check the given year is leap year or not.

Set serveroutput on

declare

year number;

begin

year := &year;

if mod (year, 4) = 0

AND

mod (year, 100) != 0

OR

mod (year, 400) = 0 then

dbms_output.put_line ('year is leap year');

else

dbms_output.put_line ('year is not leap year');

end if;

end;

/

8) Find the sum of digits of given number.

Set sumoutput on
declare

num int := 0;

i int;

s int := 0;

x int;

begin

num := #

while num num > 0 loop

x := MOD (num, 10);

s := s + x;

num := FLOOR (num / 10);

end loop;

dbms_output.put_line ('the sum of digits is' || s);

end;

/

q) check the number of vowels and consonants in a given string.

Set screenoutput on
declare

v varchar(400) := 'Joy Mohun bagan';

no. of vowels number := 0;

no. of consonants number := 0;

c char;

begin

for i in 1....length(v) loop

c := substr(v, i, 1);

if c in ('A', 'E', 'I', 'O', 'V')

OR c in ('a', 'e', 'i', 'o', 'u') then

no. of vowels := no. of vowels + 1;

else

if c not in (' ') then

no. of consonants := no. of consonants + 1;

end if;

end if;

end loop;

dbms_output.put_line('No. of vowels: || no. of vowels');

dbms_output.put_line('Count of each digit & in a number are: || count even);

end;

no. of consonants

10) count odd and even digits in a number.

Set serveroutput on

declare

 digits number := 23146579;

 length varchar2(50);

 count_odd number(10) := 0;

 count_even number(10) := 0;

begin

 for i in 1....length(digits)

 loop

 length := substr(digits, i, 1);

 if mod(length, 2) != 0 then

 count_odd := count_odd + 1;

 else

 count_even := count_even + 1;

 end if;

end loop;

 dbms_output.put_line('count of odd digits in a
 number are:' || count_odd);

 dbms_output.put_line('count of even digits in a
 number are:' || count_even);

end;

/