

- * Insert into infodys (ID, Name, Age, Email, salary, bonus)
 Values (4, 'AKhil', 19, 'akhil@gmail.com', 45000, 3000);
- * Insert into infodys (ID, Name, Age, Email, salary, bonus)
 Values (5, 'Nishant', 21, 'nish@gmail.com', 55000, 2000);

Query 18.1 Test same value during insert in Primary Key column

Insert into infodys
 Values (2, 'Nikhil', 18, 'NIK@gmail.com', 60000, 5000);

Query 18.2 Write a Query to check null value in name column
 insert into infodys
 Values (2, '', 24, 'abc@gmail.com', 50000, 6000);

Query 18.3 Write a Query to check age less than 16
 insert into infodys
 Values (3, 'Ved', 15, 'Ved@gmail.com', 40000, 5000);

Query 18.4 Write a query to check null value in ~~salary~~^{salary} column
 insert into infodys
 Values (5, 'Nishant', 21, 'Nishant@gmail.com', 2000);

Ques. Ques. Write a query to check null value in Email column
 insert into infodys
 Values (4, 'AKhil', 19, ' ', 45000, 3000);

Aim: To Perform Queries the concept of clause by clause.

Query 19: Order by clause

Query 19.1 arranging names by alphabetical order
Select * from anand order by name asc;

Query 19.2 arranging names in reverse alphabetical order

Select * from anand order by name desc;

Query

Aim: To Perform Queries using the concept of single row predefined single row function.

Query 20.1 Capitalize the elements of a column
Select upper(name), surname, salary from anand;

Query 20.2: Capitalize only the first letter of name
Select initcap(name), surname from anand;

Query 20.3: Find the length of names

Select length(name), surname from anand;

Query 20.4: Fetch Substring in clause

Select substr('abolelli', 3, 4) from user1;

Query 20.5: Fetch substring from name column

Select substr(name, 1, 4) from anand;

Results Explor Descriva Saved SQL History						
ID	NAME	FURNAME	AGE	SALARY	BOJ	DEPARTMENT
1	Tom	Smith	25	6000	01-AUG-78	HR
2	Tom	Ford	35	8000	20-JUL-85	IT
3	Donald	Smith	35	6000	05-NOV-79	HR
4	Pepe	Gold	35	6000	24-MAR-82	IT
5	Pepe	Orland	29	6000	14-JAN-80	IT
6	Pepe	Gold	35	6000	25-SEP-79	HR
7	Nikola	Gold	35	6000	21-APR-80	HR
8	Nikola	Gold	35	6000	05-JUN-80	HR

8 rows returned in 0.00 seconds

Collapse

Results Explor Descriva Saved SQL History	
INITCAP(NAME)	SURNAME
Tom	Smith
Tom	Ford
Donald	Smith
Pepe	Orland
Pepe	Gold
Nikola	Gold

8 rows returned in 0.00 seconds

Collapse

Results Explor Descriva Saved SQL History	
UPPERNAME	SURNAME
THOMAS	SMITH
THOMAS	SMITH
THOMAS	FORD
THOMAS	SMITH
THOMAS	SMITH
THOMAS	ORLAND
THOMAS	SMITH
THOMAS	SMITH
THOMAS	SMITH

8 rows returned in 0.00 seconds

Collapse

Q20.6 Query: Fetch last three character of name
 Select SUBSTR(name, -3) from anand;

Query 20.7 Use of LPad and RPad
 Select LPAD(salary, 10, 'x') from anand;
 Select RPAD(salary, 10, 'x') from anand;

Query 21.1: Use alter command to add date.
 Alter table anand add day date.

Update anand set day='10-April-20'. where ID=1

Query 21.2: Find date difference/time duration
 select name, salary, month - between (sysdate, Day)
 as "table month spent" from anand;

Aim: To perform queries using the concept of
 aggregate function
 Aggregate function/multiple row/group function

- * Multiple input and single output
- * In - max(), min(), avg(), count(); sum()
- * Create an set of row to give result per group
- * Unlike single row function, they will take.

Query 22.1 Update Anand set salary=9000
 where id=9

Query 22.2 Find maximum salary
 Select MAX(salary) as "maximumsalary" from anand;

Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History	
NAME SALARY		NAME SALARY		NAME SALARY	
Nolan 50000		Total Monthly Spend		MAXIMUM SALARY	
50000 25 8466174116420576131621744324976131		1 row(s) updated.		80000	
Riley 50000		25 8466174116420576131621744324976131422		1 rows returned in 0.00 seconds	
50000 26 251539162813360112916367956141499		0.00 seconds		277162	
Tina 50000		21 8477204605112871200610641529712			
50000 18 8299771314217443249761314277413247					
Shawn 50000		20 6464482445487394852641542738316			
50000 26 32 867790611091517986121991636					
Percy 50000		22 8279233994244548623916264154992			
50000 22 8279233994244548623916264154992					

Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History	
Maximum Salary		Total Salary		Average Salary	
30000		30000		53250	
1 rows returned in 0.00 seconds		1 rows returned in 0.00 seconds		1 rows returned in 0.00 seconds	
CSV Export		CSV Export		CSV Export	

Query 22.3 Find minimum salary
Select min(salary) as "minSal" from anand;

Query 22.31 Find the average salary
Select avg(salary) as "Average" from anand;

Query 22.5 Find the total sum of salaries
Select sum(salary) as "Total" from anand;

Query 22.6 Find the total count of salary
Select count(salary) as "Count" from anand;

Query 22.7 Find max, min and total salary
Select max(salary) as "maximumSal";
min(salary) as "minSal";
sum(salary) as "Total";
from anand;

Query : What is the different b/w Count(
and Count(column))
Select Count(*) as "Count", Count(salary) as
"Sal.Count" from anand;

What
Hence
Group By Order By
? ?

Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History	
SELECT ENAMEL FROM EMP	EMP.SALARY = 1000	SELECT SALARY FROM EMP	EMP.SALARY = 1000	SELECT SALARY FROM EMP	EMP.SALARY = 1000
1 rows returned in 0.01 seconds	0 rows returned in 0.00 seconds	1 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	1 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds
0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds
0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds

Results Explain Describe Saved SQL History		Results Explain Describe Saved SQL History	
COUNT	MAXIMUM	MINIMUM	COUNT
6	3000	2000	6
1 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds

Results Explain Describe Saved SQL History	
COUNT	Salary Count
0	0
1 rows returned in 0.00 seconds	0 rows returned in 0.00 seconds

Query: Group By function

Select Department, Sum(Salary) As "Avg" From Employee group By Department.

Query:

Select Department, Avg(Salary) As "Avg" Group By Department

Query:

Select Department, Max(Salary) As "Max" From Employee Group by Department.

Query:

Select Department, Count(Salary) As "Count" From Employee group by Department.

Query: Use of Having

~~Select Department, sum(Salary) As "Sum"~~ From Employee group By Department Having Department = "IT"

Query:

Select Department, sum(Salary) "Sum" From Employee Group By Department Having Department = "IT"

Query: Double Grouping:

Select Department - number, Department, sum(Salary) "Sum" From Employee group by Department, Number, Department;

Date _____

Expt. No. _____

Page No. _____

Query 2:

Select Department_Number, Department, sum(Salary)
As "sum" From Employee group By Department_Number
Department Having Department_Number = 100

Query 3

Select Department_Number, Department, sum(Salary)
As "sum" From Employee group By Department_Number
-number Department Having Department_Number IN
(200, 300)

Query 4:

Select Department_Number, Department, sum
(Salary) As "sum" From Employee group By
Department_Number, Department Having Department
Number in (200, 300) Order by (Department_Number)
DESC

Teacher's Signature _____

STUDY of Foreign in SQL

Query: Create table TCS_Country with a Primary Key.

Create Table TCS_Country

C.ID Number(4) Constraint C.ID PK Primary Key
) C.Name Varchar(20)

Query: Create table TCS_Location with a Primary key and connect it with TCS_Country by introducing foreign key.

Create Table TCS_Location

Loc_ID Number(4) Constraint Loc-ID-PK Primary Key
Loc_name Number(4)
City Varchar(20)

⇒ Alter Table TCS_Location

ADD C.ID Number(4)

⇒ Alter Table TCS_Location

⇒ ADD constraint C-ID-FK & TCS_Location foreign key
(C-ID) reference - TCS_Country (C.ID)

Queries

Create Table TCS_Department and introduce Primary & foreign key

Create Table TCS_Department

Deft-ID Number(4),
Deft-name Varchar(20)

Constraint Dept-ID-PK Primary Key (Dept-ID)
 Constraint Loc-ID-FK-TCS-Department Foreign Key
 (Loc-ID) reference TCS-location (Loc-ID)
);

Query: Create table TCS-Employee and introduce primary & foreign key

Create table TCS-Employee

Emp-ID number(4)
 Emp-name varchar(40)
 Emp-age number(2)
 Emp-salary number(5)
 Emp-Dob Date

Constraint emp-ID-PK Primary key (emp-ID)
 Constraint Dept-ID-FK-TCS-employee Foreign Key
 Dept-ID reference TCS-Department (Dept-ID)

Study How to Join table with the help of foreign keys.

Query: Fetch Empname, emp-salary, emp-Dob, Dept-name from DB with the help of equijoin
 Select emp-name, Emp-salary, emp-Dob, Dept-name, from TCS-Employee, TCS-Department
 where TCS-Employee, Dept-ID = TCS-Department
 Dept-ID

Query: Arrange the Previous output by Emp-name
 Select Emp-name , Emp-salary , Emp-DoJ , Dept-name
 From TCS-employee , TCS-Department.
 Where TCS-employee . dept-ID = TCS-Department . dept-ID
 order by emp-name

Query: Fetch emp-name , emp-salary , emp-doJ , dept-name from TCS-employee , TCS-Department with only IT Department

Select emp-name , emp-salary , emp-doJ , dept-name from TCS-employee , TCS-Department where
 TCS-employee dept ID = TCS-Department . dept-ID
 And TCS-Department . dept-name = 'IT'

Order by emp-name

Query: Cartesian join all the rows with each other in table

Select * From TCS-employee , TCS-Department

Query: Fetch details of employee who work in Bangkok
 Select emp-name , city from TCS-employee , TCS-Department , TCS-Location
 where TCS-Department . dept-ID = TCS-employee . dept-ID
 And TCS-Location . loc-ID = TCS-Department . loc-ID
 AND TCS-Location . city = "Bangkok"

Query: Fetch details where salary > 30000 and employee work in Bangkok

Select emp-name , city from TCS-employee , TCS-Department , TCS-Location

Date _____

Expt. No. _____

Page No. _____

Where TCS_Department-dept-ID = TCS_employee
dept = ID And TCS_location.loc-ID = TCS_department
-loc-ID And TCS_location.city = "Bangkok" And
emp_salary > 30000

Order : Order by emp-name

Select emp-name, city from TCS_location TCS-depart
-ment TCS_location,

where TCS_location.Dept-ID = TCS_employee.dept
-ID

AND TCS_location.city = "Bangkok" and emp_salary
> 30000

AND TCS_department.loc-ID = TCS_location.loc-ID

~~order~~ AND TCS_location.city = "Bangkok" AND TCS_employee
-salary emp_salary > 30000

Query : Fetch emp-name who works in Russia

Select emp-name from TCS_employee, TCS-depart
-ment, TCS_location, TCS_country

where TCS_employee.Dept-ID = TCS_department
.dept-ID ~~AND TCS~~

AND TCS_department.loc-ID = TCS_location.loc-ID

AND TCS_location.C-ID = TCS_country.C-ID

AND C-name = "Russia"

Query : Fetch in which city TCS have office in India

Select city from TCS_location, TCS_country

where TCS_location.C-ID = TCS_country.C-ID

AND TCS_country.C-name = "India".

Teacher's Signature _____

Left Query: Left outer and Right Outer Join
 Select emp-name, dept-name from TCS-employee, TCS-department where TCS-employee.dept-ID = TCS-department > Dept-ID

Select Emp-name, dept-name from TCS-employee
 left outer join TCS-Department
 on TCS-employee.dept-ID = TCS-department.dept-ID

Right outer join

Select emp-name, dept-name from TCS-employee, TCS-department
 where TCS-employee.dept-ID = TCS-department.dept-ID
 (Dept ID CT)

OR

Select emp-name, dept-name
 from TCS-employee Right outer Join TCS-department
 on TCS-employee.dept-ID = TCS-department.dept-ID

Query: Full outer join

Select emp-name, dept-name from
 TCS-employee full outer join TCS-department
 on TCS-employee.dept-ID = TCS-department.dept-ID

Query: Fetch details who is manager of which employee

Select te.emp-name As "employee-name" Tm.emp-name As "manager-name" from TCS-employee, TCS-employee Tm where te.m-ID = tm.emp-ID

Query: Fetch emp-name whose manager is arshar
select te.emp-name as "employee-name", tm.emp-name as "manager-name" from tcs-employee te, tcs-employee tm and tm.emp-name = "arshar"

Query: Fetch details where salary is b/w the low and high.

Select te.emp-name, g.grade
from tcs-employee te, grades g
where te.emp-salary between g.low and g.high

Query: Fetch grades where grades = A

Select te.emp-name, g.grade
from tcs-employee te, grades g
where te.emp-salary between g.low and g.high

Query: Fetch details where emp-salary i.e., greater than Kuldeep's salary.

Select emp-name from TCS-employee where emp-salary > (select emp-salary from TCS-employee where emp-name = "Kuldeep")

Date _____

Expt. No. _____

Page No. _____

Query: Fetch emp-name who works in IT department

Select emp-name from tcs-employee where Dept-ID =
(Select dept-ID from tc-employee where emp-name
= 'Aditya')

Query: Fetch emp-name who works in IT Department
and his salary is greater than Kuldeep.

Select emp-name from TCS-employee where
dept-ID = (Select dept-ID from TCS-employee
where emp-name = 'Aditya') and emp-salary > (Select
emp-salary from TCS-employee where emp-name
= 'Kuldeep').

Query : Fetch emp-name where salary is greater
than any salary IT department

Select emp-name from TCS-employee where emp-salary
> (Select avg(emp-salary) from TCS-employee
where dept-ID > (Select dept-ID from TCS-employee
where dept-name = 'IT'))

Teacher's Signature _____