

Employee Table

DATE
PAGE 132

E-Id	E-Name	E-Addl	E-Sal
1	Ansh	Bhopal	10,000
2	Shubh	Indore	20,000
3	Golu	Ujjain	16,000
4	Vicky	Ratlam	14,000
5	Ram	Jaipur	17,500
6	Prachi	Ajmer	20,500

① find 2nd MAX SALARY ?

→ Select * from Employee where Salary =
(Select Max(E-Sal) from Employee where
E-Sal < (Select Max(E-Sal) from employee));

② Print Id from 2 to 5 id

→ Select * from Employee where E-Id >= 2 AND
E-Id <= 5 ;

③ Print Record Starts from 'a' ?

→ Select * from Employee where E-Name like
'a%' ;

4) Print Record ends with a ?

→ Select * from Employee where E-Name like
'%a' ;

5) Print All Record in which employee Salary is
16,000 , 20,500 , 14,000 , 10,000

→ Select * from Employee where E-Sal In
(16,000 , 20,500 , 14000 , 10,000) ;

6) Create a New table with the help of existing
table and Structure

→ Create Table emp-1 as Select * from Employee;

7) find all employee Records whose Salary b/w
10,000 to 16,000 ?

Ans Select * from Employee where E-Sal between
10,000 AND 16,000

or

→ Select * from Employee where E-Sal >= 10,000
AND E-Sal <= 16,000 ;

8) Write a Statement to delete an Record ?

delete from Employee where E-Id = 3 ;

9) Write a Statement to delete more than One record in Single query

→ delete from Employee where E-Id In (3,5,6) ;

10) Insert Record into Employee table ?

→ INSERT into Employee (E-Id, E-Name, E-Add, E-Sal)
Values (7, "Suresh", "Jamshehpur", 21,000) ;

11) Update Record name of E-Id = 1

→ Update Employee Set E-name = "Anshul" where
E-Id = 1 ;

12) Create Table Query

A → Create Table Employee (E-Id int Primary Key,

E-name Varchar(10), E-Add Varchar(50),
 E-Sal Int(30)) ;

13) Foreign Key Concept

Department Table

d-Id	d-name	e-Id
1	IT	2
2	Mech	1
3	EC	3
4	Civil	5

→ Create Table Department (d-Id int Primary Key, d-name Varchar(20), e-Id int Foreign Key (E-Id) References Employee(E-Id))

Or (Cascading Concept)

→ Create Table Department (d-Id int Primary Key, d-name Varchar(20), e-Id int Foreign Key (E-Id) References Employee(E-Id) On delete Cascade On Update Cascade);

14) Query for Inner Join b/w employee table and Department Table

- Select * from Employee e JOIN department d
On e.E-ID = d.E-ID ;
- Select e.E-name , e.E-Sal , d.d-name from Employee
Inner JOIN department d
On e.E-ID = d.E-ID ;

15) Left Join Query b/w employee and department table

- Select * from Employee e
Left JOIN Department d
On e.E-ID = d.E-ID ;
- Select e.E-name , e.E-Sal , d.d-name from Employee e
Left JOIN Department d
On e.E-ID = d.E-ID ;

17) Right Join query b/w employee and department Table

- Select * from Employee e

Right Join Department d
On e.E-Id = d.E-Id ;

or

→ Select e.E-name , e.E-Sal , d.d-name from
Employee e

Right Join Department d
On e.E-Id = d.E-Id ;

18) Unmatched data (Outer Join) b/w employee and
department Table

→ (Select * from Employee e
Left Join Department d
On e.E-Id = d.E-Id
Where e.E-Id is null) Union

(Select * from Employee e
Right Join Department d
On e.E-Id = d.E-Id
Where d.E-Id is null)

19) Full Outer JOIN b/w employ and Department Table

→ (Select * from Employee e
Left Join Department d
On e.E-ID = d.E-ID)

Union

(Select * from Employee e
Right JOIN Department d
On e.E-ID = d.E-ID)

20) 5th highest Salary from Employee Table

→ Select ~~Sal~~ E-Sal from Employee Order by E-Sal
Limit 4, 1 ;

21) 5th lowest Salary from Employee Table

Select E-Sal from Employee Order by E-Sal
Limit 1 offset 4

22) Count a no. of repeated salary of group by

→ Select ~~Salary~~, E-Sal, Count(E-Sal) from Employee
group by E-Sal ;

21) 5 highest salary from Employee Table

→ Select * from Employee
Order by e-sal desc
Limit 5 ;

22) Salary between 10,000 to 20,000 Using
'Between' and Using Condition

Using Between

→ Select * from Employee where e-sal
Between 10,000 AND 20,000 ;

Using Condition

→ Select * from Employee where
e-sal >= 10,000 AND
e-sal <= 20,000 ;

23) JOIN Two Table employee and department
and Print salary b/w 10,000 to 20,000

→ Select e.e-name, e.e-sal, d.d-name from
Employee e JOIN Department d
On e.e-Id = d.c-e-Id
where e.e-sal Between 10,000 AND 20,000 ;

On

Select e.E-name, e.E-sal, d.D-name from Employee e
 JOIN Department d
 ON e.E-ID = d.D-ID
 where e.E-Sal >= 10,000 AND
 e.E-Sal <= 20,000 ;

24) Create Index On employee Table in Salary column

→ Create Index index-Salary
 On Employee (E-Sal) ;

25) Query for Distinct record of Salary in Employee Table

→ Select distinct E-Sal from employee

26) Add Column in table

→ Alter Table Employee add (mobile int)

27) Modify Column in table

→ Alter Table Employee Modify Mobile
 Varchar (10) ;

28) Delete Column from Table

→ Alter Table Employee drop mobile ;

29) Join Three Tables Query Joint Employee Table , Department Table with Project Table .

→ Create Table Project (P-Id int Primary Key ,
P-name Varchar(10) , E-Id Foreign Key
(E-Id) references Employee (E-Id)) ;

PROJECT Table :-

P-Id	P-name	E-Id
1	BanBing	3
2	Egov	1
3	hospital-many	4
4	ORS	2

Select * from (Employee e Join Department d
on e.E-Id = d.E-Id) Join Project P
on P.E-Id = e.E-Id) ;

30) Self JOIN b/w Employee and Department Table

→ Select * from Employee e1, Employee e2
where e1.E-ID = e2.E-ID ;

31) Equi JOIN b/w Employee and Department Table

→ Select * from Employee e, Department d
where e.E-ID = d.E-ID ;

32) Create View for Employee Table

→ Create View emp-View

As

Select E-name, E-Sal
from
Employee ;

Select * from emp-View ;

3) TRIGGER Example

Create table Passenger (Name Varchar(20),
 id int(10), address Varchar(20), charges int(10),
 Primary Key (id)) ;

Insert Into Passenger Values ("Ansh", 1, "Bhopal",
 10,000) ;

Insert into Passenger Values ("Shubh", 2, "Indore", 11,000);

Insert into Passenger Values ("Neeraj", 3, "Ujjain", 15,000);

Select * from Passenger

Name	id	address	charges
Ansh	1	Bhopal	10000
Shubh	2	Indore	11,000
Neeraj	3	Ujjain	15,000
PASSENGER TABLE			

Create TRIGGER flight
 before Insert

On Passenger
 for each row

Set new.charges = new.charges - 2000 ;

insert into passenger values ("Amit", 4,
"Jaipur", 10000);

Select * from Passenger

name	id	address	charges
Amit	4	Jaipur	8000

THEORY JDBC and SQL

1) What is JDBC?

Ans JDBC API is a Set of interfaces to access RDBMS is Called JDBC API

The JDBC API is a Set of classes and Interfaces . It is Used to access tabular data stored in a RDBMS (Relational Database Management System).

2) What is RDBMS ?

An application Can not be developed without a database . The database is Used to keep User data into tabular format . A table consist of Rows and Columns . Tools like