page2image568

SQL Training

Lesson-End Project Solution



**Payroll Calculation**

1. Write a query to create the **employee** and **department** tables

**SQL code: Employee table**

CREATE TABLE lep\_7.employee (

emp\_id int NOT NULL,

f\_name varchar(45) NULL,

l\_name varchar(45) NOT NULL,

job\_id varchar(45) NOT NULL,

salary decimal(8,2) NOT NULL,

manager\_id int NOT NULL,

dept\_id varchar(45) NOT NULL,

PRIMARY KEY(emp\_id));

**SQL code: Department table**

CREATE TABLE lep\_7.department (

dept\_id int NOT NULL,

dept\_name varchar(45) NOT NULL,

location varchar(45) NULL,

manager\_id varchar(45) NULL,

PRIMARY KEY(dept\_id));

1. Write a query to insert values into the **employee** and **department** tables

**SQL code: Employee table**

INSERT INSERT INTO lep\_7. employee (emp\_id,f\_name,l\_name,job\_id,salary,manager\_id,dept\_id) VALUES ('103','krishna','gee','HP125','500000','05','44')

**SQL code: Department table**

INSERT INTO lep\_7. department (dept\_id,dept\_name,location,manager\_id) VALUES ('24','production','india','2');

1. Write a query to create a view of the **employee** and **department** tables

**SQL code:**

CREATE VIEW emp AS SELECT f\_name,l\_name,salary ,dept\_name,location,emp\_id FROM lep\_7.employee,lep\_7.department WHERE l\_name = 'jain';

1. Write a query to display the first and last names of every employee in the **employee** table whose salary is greater than the average salary of the employees listed in the **SQL basics** table

**SQL code:**

SELECT e.f\_name,e.l\_name

FROM lep\_7.employee e,sqlbasics.emp s

WHERE e.salary > s.salary;

**Output:**

Table

Description automatically generated

1. Write a query to change the delimiter to //

**SQL code:**

delimiter //

1. Write a query to create a stored procedure in the **employe**e table for every employee whose salary is greater than or equal to 250,000

**SQL code:**

use lep\_7;

SELECT \* from employee;

delimiter &&

CREATE PROCEDURE top\_salarys()

BEGIN

SELECT job\_id,f\_name,salary

FROM employee where salary>=250000;

END &&

delimiter ;;

1. Write a query to execute the stored procedure

**SQL code:**

call top\_salarys();

**Output:**

Table

Description automatically generated

1. Write a query to create and execute a stored procedure with one parameter using the **order by** functionin descending order of the salary earned

**SQL code:**

delimiter //

CREATE PROCEDURE sort\_salarys(IN var INT)

BEGIN

SELECT job\_id,f\_name,salary

FROM employee ORDER BY salary DESC LIMIT var;

end //

delimiter ;

call sort\_salarys(3);

**Output:**

Table

Description automatically generated