DBMS Evaluation Paper:

1. Display the Employee's ID, Column with the concatenation of First Name and Last Name, Salary who is living in 'London'.
->
```
select E.employee_id as "Employee Id" , concat(E.first_name,' ',E.last_name) as "Employee full name",E.salary as "Employee salary" from employees E
inner join departments D
on E.department_id = D.department_id
inner join locations L
on D.location_id = L.location_id
where L.city = 'London';
```

2. Display the Employee's Last name whose name starts with 's'.
->
```
select last_name as "Employee Last Name" from employees where first_name like 'S%';
```

3. Find the Employee Joining date on the first Monday in October month.
->
```
Select employee_id, hire_date as 'Employee Joining Date' from employees where month(hire_date)=10 and day(hire_date) in (1,2,3,4,5,6,7) and dayname(hire_date)='monday' order by hire_date;
```

4. Display the Employee name and city in capital letter.
->
```
select UPPER(concat(E.first_name,' ',E.last_name)) as "Employee Full Name",UPPER(L.city) as "Employee City" from employees E
inner join departments D
on E.department_id = D.department_id
inner join locations L
on D.location_id = L.location_id ;
```

5. Display the department and their max salary of the same department.
->
```
select D.department_name, max(E.salary) from employees E left join departments D on E.department_id=D.department_id group by E.department_id;
```

6. Find the second-lowest salary among the employees, who are working in the same department.
->
```
select distinct salary, department_id FROM employees e1 where 2 = (select count(distinct salary) from employees e2 where e2.salary <= e1.salary and e1.department_id= e2.department_id) ;
```

7. Create a report that has employee name, department name, job id, manager name have a salary in range 10000 to 25000 and department = 90.
->
```
select concat(E.first_name,' ',E.last_name) as "Employee full name" , D.department_name as "Employee department name" , E.job_id as "Employee job Id" , concat(F.first_name,' ',F.last_name) as "Manager name" from employees E
inner join departments D
on E.department_id = D.department_id
left join employees F
on E.manager_id = F.employee_id
where E.salary >= 10000 and E.salary <= 25000 and E.department_id = 90;
```

8. Display job title, employee name, and the difference between the salary of the employee and the minimum salary for the job.
->
```
select J.job_title as "Employee Job Title" , concat(E.first_name,' ',E.last_name) as "Employee full name", (E.salary - (select min(salary) from employees)) as "Salary Difference" from employees E
inner join jobs J
on E.job_id = J.job_id ;
```

9. Write a query to display department name, name (first_name, last_name), hire date, the salary of the manager for all managers whose experience is more than 15 years.
->
```
select D.department_name as "Manager department name",concat(E.first_name,' ',E.last_name) as "Manager full name", E.hire_date as "Hire Date", E.salary as "Manager's salary"
from employees E inner join departments D
on E.employee_id = D.manager_id
where year(curdate())-year(E.hire_date) > 15;
```

10. Create the procedure to insert and update the record in the employee table and whenever there is any update, create the logs for the same keeping old value and new value, date, and user name.
->

Procedure:
```
-------------------------------------------------------------
delimiter //
drop procedure if exists insertProc //
create procedure insertProc( empid INT, fname VARCHAR(20), lname VARCHAR(20), email VARCHAR(30), hiredate DATETIME, jobid VARCHAR(20), sal INT, depid INT)
begin
insert into
employees(employee_id, first_name, last_name, email, hire_date, job_id, salary, department_id)
values(
empid, fname , lname , email , hiredate, jobid, sal, depid
)
;
end//
delimiter ;
```

Logs Table:

```
-- -----------------------------------------------------
create table updateLogs(
logId int auto_increment primary key,
userName varchar(50) not null unique,
logDate datetime default now(),
oldValue varchar(100),
newValue varchar(100)
);
```

Trigger:

```
-- -----------------------------------------------------
delimiter //
drop trigger if exists updateLogs //
create trigger updateLogs after
update
on employees for each row
begin
declare oldValue varchar(80);
declare newValue varchar(80);
set oldValue = concat( old.employee_id ,', ', old.first_name , ', ' , old.last_name , ', ' , old.hire_date
, ', ' , old.job_id ,', ' , old.salary , ', ' , old.department_id);
set newValue = concat( new.employee_id , ', ' , new.first_name , ', ' , new.last_name , ', ' ,
new.hire_date , ', ' , new.job_id , ', ' , new.salary , ', ' , new.department_id);
insert into
updateLogs(userName, logDate, oldValue, newValue)
values
(
user() ,curdate(), oldValue, newValue
);
end//
delimiter ;
```