Sample Questions - Self Practice
(LBJ Module 1– DBMS & SQL)

Reference Tables

Departments Table		
Department_ID	Number(3)	Primary Key
Department_Name	Varchar(30)	Not Null

Employees Table		
Employee_ID	Number(3)	Primary Key
First_Name	Varchar(50)	Not Null
Last_Name	Varchar(50)	Not Null
Salary	Number(6)	Not Null
Email_ID	Varchar(50)	Unique Key
Job_ID	Varchar(10)	Foreign Key
Hire_Date	Datetime	Not Null
Department_ID	Number(3)	Foreign Key
Manager_ID	Number(3)	Foreign Key

Jobs Table		
Job_ID	Varchar(10)	Primary Key
Job_Title	Varchar(50)	Not Null

Q1. Write a query to display Employee ID, Employee Full Name, Salary, Annual Salary, Job ID , date on which employee was hired for employees working in department ID 50.

Q2. Write a query to display unique job ID's from the Employees Table

Q3. Create a query to display all the data from Employees table. Separate each column by a comma.

Q4. Display employee Id, last name and hire date of every employee who was hired in 2010.

Q5. Display employee Id, last name and hire date of every employee who do not have manager.

Q6. Display employee Id, last name, job ID and hire date of all employees where the third letter of last name is an -- a.

Q7. Display employee Id, last name, job ID and hire date of all employees where the third letter of last name is an – a and e.

Q8. Display employee Id, last name, job ID, Salary and hire date of all employees who works in department ID 20, 50 and 80.

Q9. Display employee Id, last name, job ID, Salary and hire date of all employees whose salary ranges from 5000 to 15000.
Q10. Write a query to display employee Id, last name, Salary, department ID and department name for all the employees working in department ID 50.

Q10. Write a query to display employee Id, last name, Salary, job id, department ID, job title and department name for all the employees working in department ID 100.

Q11. Display the name and hire date for all the employees who were hired before their managers, along with manager's name and hire date.
Label the columns: **Employee, EmpHired, Manager, MgrHired**

Q12. Determine the validity of the following three statements. Highlight either True or False

1. Group functions works across many rows to produce one result per group.
   True / False

2. Group functions include nulls in calculations.
   True / False

3. The Where Clause restricts rows prior to inclusion in a group calculation.
   True / False

Q13. Display Highest, lowest, sum and average salary of all the employees. Label the columns Maximum, Minimum, Sum and Average respectively.

Q14. Display Department ID, Highest, lowest, sum and average salary of all the employees for each department.

Q15. Display Job ID, Highest, lowest, sum and average salary of all the employees for each job type.

Q16. Write a query to display the number of people with the same job.

Q16. Write a query to display the number of people working in same department.

Q17. Write a query to display Department name, number of employees and average salary of all employees in that department.

Q18. Write a query to display employee ID, last name and salary of all the employees who earns more than avg. salary.

Q19. Write a query to display employee ID, last name and hire date of any employee in the same department as Jack. Exclude Jack.

Q20. Select the appropriate answers: (Additional Challenge)

a.) You need to create a procedure named INSERT_LOC that will insert a new location into the LOCATIONS table. Identify the correct procedure code to perform the preceding task?

i) CREATE PROCEDURE INSERT_LOC(p_lid NUMBER, p_city VARCHAR2)
IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('PROCEDURE STARTED');
  INSERT INTO LOCATIONS (location_id, city) VALUES (p_lid, p_city);
END;

ii) CREATE PROCEDURE INSERT_LOC
IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('PROCEDURE STARTED');
  INSERT INTO LOCATIONS(location_id,city) VALUES(p_lid,p_city);
END;

iii) CREATE PROCEDURE INSERT_LOC(p_lid NUMBER, p_city VARCHAR2)
IS
  DBMS_OUTPUT.PUT_LINE('PROCEDURE STARTED');
  INSERT INTO LOCATIONS(location_id,city) VALUES(p_lid,p_city);
END;

iv) CREATE PROCEDURE INSERT_LOC(p_lid NUMBER, p_city VARCHAR2)
IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('PROCEDURE STARTED');
  INSERT INTO LOCATIONS(location_id,city) VALUES(id,city);
END;

b.) You need to create a procedure named INSERT_DEPT that will insert a new department details into the DEPARTMENTS table. Identify the correct procedure code to perform the preceding task?

i) CREATE PROCEDURE INSERT_DEPT(id NUMBER, name VARCHAR2)
IS
BEGIN
  INSERT INTO DEPARTMENTS VALUES(id,name);
END;

ii) CREATE PROCEDURE INSERT_DEPT(id NUMBER, name VARCHAR2)
IS
BEGIN

```
   INSERT INTO DEPARTMENTS VALUE(id,name);
END;
```

iii)
```
CREATE PROCEDURE INSERT_DEPT
IS
BEGIN
   INSERT INTO DEPARTMENTS VALUES(id,name);
END;
```

iv)
```
CREATE PROCEDURE INSERT_DEPT(id NUMBER, name VARCHAR2)
IS
BEGIN
   INSERT INTO DEPARTMENTS VALUES(dept_id,dept_name);
END;
```