

## DATABASE MANAGEMENT SYSTEM LAB MANUAL

---

**ASSIGNED BY: Madan Lal**

**ASSIGNED TO: Sir, Khalid Hussain**

**CMS-ID: 053-18-0005**

**SECTION: A**

## **Lab 01**

- Q1: SELECT FIRST\_NAME AS 'First Name', LAST\_NAME AS 'Last Name' FROM employees;
- Q2: SELECT DISTINCT DEPARTMENT\_ID FROM employees;
- Q3: SELECT \* FROM employees ORDER BY FIRST\_NAME DESC;
- Q4: SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME FROM EMPLOYEES ORDER BY SALARY ASC;
- Q5: SELECT SUM(SALARY) FROM employees;
- Q6: SELECT MIN(SALARY),MAX(SALARY) FROM employees;
- Q7: SELECT COUNT(SALARY),AVG(SALARY) FROM employees;
- Q8: SELECT COUNT(DISTINCT JOB\_ID) FROM EMPLOYEES;
- Q9: SELECT UPPER(FIRST\_NAME) FROM employees;
- Q10: SELECT \* FROM EMPLOYEES LIMIT 10;
- Q11: SELECT \* FROM EMPLOYEES LIMIT 3,2;
- Q12: SELECT \* FROM EMPLOYEES ORDER BY EMPLOYEE\_ID DESC LIMIT 1,1;

## **Lab 02**

- 1.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees WHERE SALARY < 3000;
- 2.SELECT FIRST\_NAME, LAST\_NAME FROM employees WHERE FIRST\_NAME LIKE 'A%';
- 3.SELECT FIRST\_NAME, JOB\_ID, DEPARTMENT\_ID FROM employees WHERE JOB\_ID = 'PU\_CLERK' OR  
MANAGER\_ID =114;
- 4.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees WHERE SALARY >= 1500 AND  
SALARY <= 3000;
- 5.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees WHERE COMMISSION\_PCT = 0;
- 6.SELECT FIRST\_NAME FROM employees WHERE FIRST\_NAME LIKE '%N';
- 7.SELECT FIRST\_NAME, JOB\_ID, DEPARTMENT\_ID FROM employees WHERE JOB\_ID != 'PU\_CLERK';
- 8.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees WHERE NOT SALARY  
IN(3300,3200,2200);
- 9.SELECT FIRST\_NAME FROM employees WHERE FIRST\_NAME LIKE 'A%N';
- 10.SELECT FIRST\_NAME FROM employees WHERE FIRST\_NAME LIKE '%LA%';
- 11.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees ORDER BY SALARY DESC;

- 12.SELECT FIRST\_NAME FROM employees WHERE FIRST\_NAME LIKE '\_a%';
- 13.SELECT EMPLOYEE\_ID, FIRST\_NAME, SALARY FROM employees WHERE SALARY NOT BETWEEN 1500 AND 3000;
- 14.SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM employees WHERE DEPARTMENT\_ID = 30 OR DEPARTMENT\_ID = 100 ORDER BY DEPARTMENT\_ID ASC;
- 15.SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE NOT (SALARY BETWEEN 10000 AND 15000) AND (DEPARTMENT\_ID <> 30 AND DEPARTMENT\_ID <> 100);
- 16.SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM employees WHERE YEAR(HIRE\_DATE) = 1987;
- 17.SELECT LAST\_NAME FROM employees WHERE LAST\_NAME LIKE '\_\_\_\_';
- 18.SELECT FIRST\_NAME, SALARY, SALARY\*0.15 FROM EMPLOYEES;
- 19.SELECT FIRST\_NAME, SALARY, COMMISSION\_PCT\*SALARY FROM EMPLOYEES;
- 20.SELECT FIRST\_NAME, SALARY, SALARY-500 FROM EMPLOYEES;

## **Lab 03**

- Q1: SELECT CONCAT(LEFT(FIRST\_NAME,3),RIGHT(JOB\_ID,3)) FROM employees;
- Q2: SELECT REPLACE(JOB\_ID,'E','A') FROM employees;
- Q3: SELECT FIRST\_NAME, YEAR(HIRE\_DATE) FROM employees;
- Q4: SELECT FIRST\_NAME, DATE\_FORMAT(HIRE\_DATE, '%M%d%Y') FROM EMPLOYEES WHERE HIRE\_DATE < "1987-7-30";
- Q5: SELECT ADDDATE(CURRENT\_DATE(), INTERVAL -3 MONTH);
- Q6: SELECT DAY(CURRENT\_DATE());
- Q7: SELECT DATE\_FORMAT(CURRENT\_DATE(), '%M %d, %Y');
- Q8: SELECT DATE\_FORMAT(CURRENT\_DATE(), '%W %M %Y');
- Q9: SELECT FIRST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE HIRE\_DATE BETWEEN '1987-06-01' AND '1987-7-30';
- Q10: SELECT DATE\_FORMAT(CURRENT\_DATE(), '%d/%m/%Y');
- Q11: SELECT FIRST\_NAME, LAST\_NAME FROM employees WHERE MONTH(HIRE\_DATE) LIKE '7';
- Q12: SELECT CONCAT(EMAIL, '@iba-suk.edu.pk') FROM employees;
- Q13: SELECT EMPLOYEE\_ID, FIRST\_NAME, MID(HIRE\_DATE, 6, 2) FROM employees;
- Q14: SELECT RIGHT(PHONE\_NUMBER, 4) FROM employees;

Q15: SELECT RIGHT(STREET\_ADDRESS, INSTR(REVERSE(STREET\_ADDRESS), ' ')) FROM locations;

Q16: SELECT MIN(STREET\_ADDRESS) FROM locations;

Q17: SELECT LEFT(JOB\_TITLE, INSTR(JOB\_TITLE, ' ')) FROM JOBS;

Q18: SELECT LENGTH(FIRST\_NAME) FROM employees WHERE LAST\_NAME LIKE '\_C%';

## **Lab 04**

Q1: SELECT COUNT(\*), DEPARTMENT\_ID FROM EMPLOYEES GROUP BY DEPARTMENT\_ID;

Q2: SELECT DEPARTMENT\_ID FROM EMPLOYEES GROUP BY DEPARTMENT\_ID HAVING COUNT(DEPARTMENT\_ID) >= 5;

Q3: SELECT \* FROM EMPLOYEES GROUP BY FIRST\_NAME;

Q4: SELECT FIRST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP '^....\$';

Q5: SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP 'LA' OR LAST\_NAME REGEXP 'LA';

Q6: SELECT FIRST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP '^A[A-Z]\*N\$';

Q7: SELECT FIRST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP '^A[A-Z]\*N\$';

Q8: SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP '^A[A-Z]\*\$';

Q9: SELECT COUNT(\*) FROM EMPLOYEES GROUP BY JOB\_ID;

Q10: SELECT MANAGER\_ID, MIN(SALARY) FROM EMPLOYEES WHERE MANAGER\_ID <> 0 AND SALARY > 2000 ORDER BY SALARY DESC;

Q11: SELECT COUNT(\*) FROM EMPLOYEES WHERE COMMISSION\_PCT = 0;

Q12: SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME REGEXP '^[t]';

## **Lab 05**

1. SELECT first\_name, last\_name, department\_id, department\_name FROM employees INNER JOIN departments USING(department\_id);

2. SELECT first\_name, last\_name, job\_id, department\_id, department\_name FROM employees JOIN departments USING(department\_id) JOIN locations USING(location\_id) WHERE city = 'London';

3. SELECT first\_name, last\_name, department\_name, city, state\_province FROM employees JOIN departments USING(department\_id) JOIN locations USING(location\_id);

4. SELECT EMPLOYEE\_ID, LAST\_NAME 'EMPLOYEE', MANAGER\_ID, LAST\_NAME 'MANAGER' FROM EMPLOYEES;

5. SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE HIRE\_DATE > (SELECT HIRE\_DATE FROM EMPLOYEES WHERE LAST\_NAME='Jones');
6. SELECT department\_name, COUNT(\*) FROM departments INNER JOIN employees USING(department\_id) GROUP BY department\_name;
7. SELECT department\_id, department\_name, first\_name 'Manager\_name' FROM departments inner join employees using(department\_id);
8. SELECT DEPARTMENT\_NAME, FIRST\_NAME 'MANAGER\_NAME', CITY FROM EMPLOYEES INNER JOIN DEPARTMENTS USING(DEPARTMENT\_ID) INNER JOIN LOCATIONS USING(LOCATION\_ID);
9. SELECT \* FROM job\_history JOIN employees USING(employee\_id) WHERE salary > 10000;
10. SELECT first\_name, last\_name, hire\_date, salary, (DATEDIFF(now(), hire\_date))/365 Experience FROM departments JOIN employees USING(manager\_id) WHERE (DATEDIFF(now(), hire\_date))/365>15;
11. SELECT department\_name, AVG(salary), COUNT(\*) FROM departments JOIN employees USING (department\_id) WHERE COMMISSION\_PCT=0 GROUP BY department\_name;
12. SELECT country\_name,city, department\_name FROM countries JOIN locations USING (country\_id) JOIN departments USING (location\_id);
13. SELECT department\_name, CONCAT(first\_name, ' ', last\_name) AS name\_of\_manager FROM departments JOIN employees USING(manager\_id);
14. SELECT \* FROM jobs JOIN employees USING(job\_id) WHERE salary >= 12000;
15. SELECT CONCAT(first\_name, ' ', last\_name) AS Employee\_name, salary FROM employees JOIN departments USING(department\_id) JOIN locations USING (location\_id) WHERE city = 'London';
16. SELECT job\_title, first\_name, salary-min\_salary FROM employees NATURAL JOIN jobs;
17. SELECT job\_title, AVG(salary) FROM employees NATURAL JOIN jobs GROUP BY job\_title;
18. SELECT employee\_id, job\_title, end\_date-start\_date Days FROM job\_history NATURAL JOIN jobs WHERE department\_id=90;

## **Lab 06**

- 1.SELECT FIRST\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE\_ID=103);
- 2.SELECT \* FROM DEPARTMENTS WHERE DEPARTMENT\_NAME='Sales' OR DEPARTMENT\_NAME='IT';
- 2.select \* from employees join departments using(department\_id) where department\_name in('Sales', 'IT');

- 3.SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY = ANY(SELECT AVG(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT\_ID);
- 4.SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE LAST\_NAME <> 'Fox' GROUP BY DEPARTMENT\_ID;
- 5.SELECT FIRST\_NAME, LAST\_NAME, EMPLOYEE\_ID FROM EMPLOYEES NATURAL JOIN LOCATIONS WHERE CITY='London';
- 6.SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME LIKE '%Z%' GROUP BY DEPARTMENT\_ID;
- 7.SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE SALARY > (SELECT SUM(SALARY)/2 FROM EMPLOYEES GROUP BY DEPARTMENT\_ID);
- 8.SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID IN(SELECT MANAGER\_ID FROM EMPLOYEES);
- 8.SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID = ANY(SELECT MANAGER\_ID FROM EMPLOYEES);
- 9.SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_NAME, CITY FROM EMPLOYEES NATURAL JOIN DEPARTMENTS NATURAL JOIN LOCATIONS WHERE SALARY > ALL(SELECT SALARY FROM EMPLOYEES NATURAL JOIN JOB\_HISTORY WHERE START\_DATE BETWEEN 1990-01-01 AND 1991-12-31);
- 10.SELECT \* FROM DEPARTMENTS WHERE DEPARTMENT\_ID NOT IN (SELECT DEPARTMENT\_ID FROM EMPLOYEES);
- 11.SELECT \* FROM JOBS NATURAL JOIN LOCATIONS WHERE JOB\_ID='ST\_CLERK' OR JOB\_ID='SA\_REP' OR JOB\_ID='AD\_ASST' AND CITY='Seattle';
- 12.SELECT \* FROM EMPLOYEES a, EMPLOYEES b WHERE a.MANAGER\_ID = b.EMPLOYEE\_ID AND a.SALARY> b.SALARY;
- 13.
- 14.SELECT \* FROM DEPARTMENTS NATURAL JOIN EMPLOYEES WHERE SALARY >= 8000;
- 15.SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) AS MANAGER\_NAME FROM EMPLOYEES WHERE EMPLOYEE\_ID IN (SELECT MANAGER\_ID FROM EMPLOYEES GROUP BY MANAGER\_ID HAVING COUNT(\*)>=4);

## **Lab 07**

1. CREATE TABLE EMPLOYEES\_REPLICA( EMPLOYEE\_ID INT(3), FIRST\_NAME VARCHAR(10), LAST\_NAME VARCHAR(10), EMAIL VARCHAR(15), PHONE\_NUMBER INT(12), HIRE\_DATE INT(15), JOB\_ID INT(10), SALARY INT(10), COMMISSION\_PCT INT(10), MANAGER\_ID INT(10), DEPARTMENT\_ID INT(10));

2. ALTER TABLE employees\_replica add(Address varchar(20));
3. ALTER TABLE employees\_replica drop Address;
4. ALTER TABLE employees\_replica add(House\_No varchar(20), Street\_No varchar(20), Area int(20), City varchar(15));
5. ALTER TABLE EMPLOYEES\_REPLICA MODIFY COLUMN HOUSE\_No INT(20);
6. CREATE DATABASE TEMP;
- 6.1 CREATE TABLE FACULTY (FACULTY\_ID INTEGER, FACULTY\_NAME VARCHAR(25) PRIMARY KEY(FACULTY\_ID));
- 6.2 CREATE TABLE COURSE (COURSE\_ID VARCHAR(8), COURSE\_NAME VARCHAR(15));
- 6.3 CREATE TABLE CLASS (CLASS\_ID VARCHAR(8), COURSE\_ID VARCHAR(8), SECTION\_NO INT(10), SEMESTER VARCHAR(10));
- 6.4 CREATE TABLE STUDENT (STUDENT\_ID INT(10), STUDENT\_NAME VARCHAR(25), FACULTY\_ID INT(10) PRIMARY KEY(STUDENT\_ID), FOREIGN KEY (FACULTY\_ID));
7. ALTER TABLE STUDENT ADD(CLASS VARCHAR(20));
8. ALTER TABLE DEPARTMENT RENAME TO DEPT;
8. RENAME TABLE DEPARTMENT TO DEPT;
9. ALTER TABLE LOCATIONS ADD(REGION\_ID INT(10));
10. ALTER TABLE LOCATIONS CHANGE COLUMN STATE\_PROVINCE STATE VARCHAR(20);

## **Lab 08**

1. create table DEPARTMENT(DEPTNO numeric Primary key, DNAME varchar(10), LOC varchar(10));
  2. create table Employee(EMPNO numeric Primary key, ENAME varchar(10), SAL numeric, DEPTNO numeric, foreign key (DEPTNO) references department (DEPTNO));
  3. Alter table Employee add check(ENAME REGEXP '[A-Z]');
  4. Alter table department modify DNAME varchar(10) NOT NULL;
  5. Alter table Employee modify SAL numeric unique;
- OR
5. ALTER TABLE EMPLOYEE ADD UNIQUE(SAL);
  6. Alter table Employee add foreign key(DEPTNO) references department(DEPTNO) ON UPDATE CASCADE ON DELETE SET NULL;

## **Lab 09**

1.

```
INSERT INTO projects (project_id, name, start_date, end_date) values(1,'AI for Marketing','2019-08-01','2019-12-29'),
```

```
(2,'ML for sales','2019-05-15','2019-11-20'),(3,'CS for IT','2020-01-01','2020-05-20'),
```

```
(4,'SQL for input','2020-06-13','2020-11-20');
```

```
INSERT INTO student (std_NO, std_Name, Department, email, phone, project_id)
```

```
values ('S100','Ali Mehmood','Admisnistration','ali@iba-suk.edu.pk','0333-895311',3),
```

```
('S101','Manisha Kataria','Computer Science','manisha@iba-suk.edu.pk','0345-111333444',2),
```

```
('S102','Sagar Sanjay','Engineering','sagar@iba-suk.edu.pk','0300-22224454',2),
```

```
('S103','Sara Shaikh','IT','sara@iba-suk.edu.pk','0300-111110000',3);
```

2. DELETE FROM projects;

3. INSERT INTO projects SELECT \* FROM project\_copy;

4. UPDATE projects SET Start\_date='2023-02-01' WHERE name='CS for IT';

5. UPDATE projects SET cost=90000 WHERE cost=NULL;

6. UPDATE student join projects USING(project\_id) SET  
daysToComplete=DATEDIFF(start\_date,end\_date);

## **Lab 10**

1.

```
DELIMITER $$
```

```
CREATE PROCEDURE DISPLAY()
```

```
BEGIN
```

```
SELECT employee_id, concat(first_name," ",last_name) as Employee_Name, salary
```

```
FROM employees
```

```
WHERE department_id=10;
```

```
END$$
```



DELIMITER ;

2.

DELIMITER \$\$

```
CREATE PROCEDURE DISPLAY2(IN DEPTNO decimal(4,0),OUT DEPTNAME varchar(30), OUT  
TotalSalary decimal(8,0))
```

```
BEGIN
```

```
SELECT department_name, sum(salary)
```

```
FROM departments join employees using(department_id);
```

```
END$$
```

DELIMITER ;

3.

DELIMITER \$\$

```
CREATE PROCEDURE DISPLAY3(IN DEPTNO decimal(4,0), OUT DNAME varchar(15), OUT  
SMALLEST_SALARY decimal(5,0),
```

```
HIGHEST_SALARY decimal(10,0))
```

```
BEGIN
```

```
SELECT employee_id, concat(first_name," ",last_name), department_name, min(salary), max(salary),  
sum(salary)
```

```
FROM employees join departments using(department_id)
```

```
WHERE department_id=DEPTNO;
```

```
END$$
```

DELIMITER ;

4.

DELIMITER //

```
CREATE FUNCTION MANAGER()
```

```
RETURNS DECIMAL(10,0)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
DECLARE EMP DECIMAL(10,0);
```

```
SELECT SUM(salary)
INTO EMP
FROM employees
WHERE employee_id=manager_id;
return EMP;
END//
```

```
DELIMITER ;
```

5.

```
DELIMITER //
CREATE FUNCTION MANAGER2(empno decimal(4,0))
RETURNS varchar(15)
DETERMINISTIC
BEGIN
DECLARE managerName varchar(15);
SELECT CONCAT(first_name," ",last_name) as managerName
INTO managerName
FROM employees
WHERE employee_id=manager_id AND employee_id=empno;
RETURN managerName;
END //
```

```
DELIMITER ;
```

6.

```
DELIMITER $$
CREATE FUNCTION MANAGER3(managerName varchar(10))
RETURNS DECIMAL(10,0)
DETERMINISTIC
BEGIN
DECLARE AVG_Salary DECIMAL(10,0);
```

```
SELECT AVG(SALARY) as AVG_SALARY
INTO AVG_SALARY
FROM employees
WHERE first_name=managerName;
RETURN AVG_SALARY;
END$$
DELIMITER ;
```

## **Lab 11**

```
1.
DELIMITER $$
CREATE PROCEDURE countEven (IN anyNumber int)
BEGIN
DECLARE count int;
DECLARE sum int;
set sum = 0;
set count = 2;

anyNumber: LOOP
    IF anyNumber < count
        THEN LEAVE anyNumber;
    ELSEIF mod(anyNumber,2) = 0
        THEN SET anyNumber = anyNumber+count;
    END IF;
    SET count=count+1;
    END LOOP anyNumber;
END$$
DELIMITER ;
```

////////////////////////////////////

2.

DELIMITER \$\$

CREATE PROCEDURE checkCustomer (IN customerNumber INT)

BEGIN

DECLARE check\_customer INT;

DECLARE isPresent varchar(20);

SELECT customerNumber

INTO check\_customer

From customers

WHERE check\_customer = customerNumber;

IF check\_customer IS NOT NULL

THEN SET isPresent="Customer Exists";

ELSE SET isPresent="Customer not Exists";

END IF;

SELECT isPresent;

END\$\$

DELIMITER ;

////////////////////////////////////

3.

DELIMITER \$\$

CREATE PROCEDURE CountEmployees (OUT count\_employees INT, IN office\_code INT)

BEGIN

CASE

WHEN office\_code>3

THEN SELECT count(\*)

INTO count\_employees

FROM employees

```
WHERE office_code > '3';
WHEN office_code<3
    THEN SELECT count(*)
    INTO count_employees
    FROM employees
    WHERE office_code < '3';
Else
    SELECT count(*)
    INTO count_employees
    FROM employees
    WHERE office_code = '3';
END CASE;
END$$
DELIMITER ;
////////////////////////////////////
4.
DELIMITER $$
CREATE PROCEDURE EvenOdd(IN anyNumber INT)
BEGIN
DECLARE check_EvenOdd varchar(20);
    IF mod(anyNumber,2)=0
        THEN SET check_EvenOdd = "Number is Even";
    ELSE
        SET check_EvenOdd = "Number is Odd";
    END IF;
SELECT check_EvenOdd;
END$$
DELIMITER ;
```

## **Lab 12**

1.

```
CREATE VIEW displayRecords AS
```

```
SELECT employee_id, CONCAT(first_name," ",last_name) as Employee_Name, salary, department_id  
FROM employees WHERE department_id=10;
```

2.

```
SELECT * FROM displayRecords;
```

3.

```
UPDATE displayRecords SET department_id=20 WHERE employee_id=200;
```

```
SELECT * FROM displayRecords;
```

```
>> >> Record with employee_id 200 is missing...
```

4.

```
CREATE VIEW Simple AS
```

```
SELECT * FROM EMPLOYEES
```

```
WITH CHECK OPTION;
```

5.

```
CREATE VIEW Display AS
```

```
SELECT CONCAT(first_name," ",last_name) as Employee_Name, department_name, salary
```

```
FROM employees join departments USING(department_id)
```

```
WHERE department_id=20;
```

6.

```
CREATE TABLE EMP55 AS
```

```
SELECT * FROM employees WHERE department_id=10;
```

```
CREATE VIEW MY_VU AS
```

```
SELECT * FROM EMP55 WHERE department_id=10;
```

7.

```
CREATE TABLE Emp2 AS
```

```
SELECT * FROM EMPLOYEES;
```

CREATE INDEX First\_name on Emp2(First\_name);

8.

CREATE INDEX department\_name on departments(department\_name);

9.

DROP INDEX first\_name on Emp2;

10.

SHOW INDEXES FROM DEPARTMENTS;

DROP INDEX `PRIMARY` ON departments;

DROP INDEX DEPT\_MGR\_FK ON departments;

DROP INDEX DEPT\_LOCATION\_IX ON departments;

DROP INDEX department\_name ON departments;

**~END OF LABS~**