SE-COMP (DIV-B)	**Roll number :** 8942
Experiment no. : 5 (part2)	**Date of Implementation :** 26/03/2021

Aim : To implement Integrity constraints

Tool Used : MySql / PostgreSQL

Related Course outcome : At the end of the course, Students will be able to Use

SQL : Standard language of relational database

Rubrics for assessment of Experiment:

Indicator	Poor	Average	Good
Timeliness • Maintains assignment deadline (3)	Assignment not done (0)	One or More than One week late (1-2)	Maintains deadline (3)
Completeness and neatness • Complete all parts of QUERY assignment(3)	N/A	< 80% complete (1-2)	100% complete (3)
Originality • Extent of plagiarism(2)	Copied it from someone else(0)	At least few questions have been done without copying(1)	Assignment has been solved completely without copying (2)
Knowledge • In depth knowledge of the QUERY assignment(2)	Unable to answer 2 questions(0)	Unable to answer 1 question (1)	Able to answer 2 questions (2)

Assessment Marks :

Timeliness(3)	
Completeness and neatness(3)	
Originality (2)	
Knowledge (2)	
Total (Out of 10)	

Remark:

Teacher's Sign :

Experiment No. 5- Integrity Constraints

AIM:

- To implement database for relational model using DDL statement
- Apply Integrity Constraints for the specified system

Objective of the Experiment:

After completing this experiment you will be able to:

- Create database.
- Create table with constraints
- Modify the schema of the table.

Theory :

Pre Lab/ Prior Concepts:

The Data Definition Language (DDL) is used to create and modify the relational schema. Also it is used to add various constraints to the table like the primary key, foreign key, check constraint, not null constraint and unique constraint.

The DDL statements are:

CREATE

DROP

ALTER

SQL supports the standard int, smallint, real, double precision, char(N), varchar(N), date, time, timestamp, and interval for creating tables.

Procedure / Algorithm:

Create Database and use it:

```
$ createdb mydb
$ psql mydb
```

Delete a database:

```
$ dropdb mydb
```

Create table:

```
CREATE TABLE my_first_table
( first_column text,
second_column integer
);
```

```
CREATE TABLE products (
product_no integer,
name text, price numeric);
```

Drop Table:

```
DROP TABLE my_first_table;
DROP TABLE products;
```

Default Value:

```
CREATE TABLE products (
product_no integer,
name text,
price numeric DEFAULT 9.99 );
```

Constraints:

1. Primary Key

```
CREATE TABLE products (
product_no integer PRIMARY KEY,
name text,
price numeric );
```

Primary keys can also constrain more than one column.

```
CREATE TABLE example (
a integer,
b integer,
c integer,
PRIMARY KEY (a, c)
);
```

2. Check Constraint

```
CREATE TABLE products (
product_no integer,
name text,
price numeric CHECK (price > 0) );
```

3. Not Null Constraint

```
CREATE TABLE products (
product_no integer NOT NULL,
name text NOT NULL,
price numeric );
```

4. Unique Constraint

```
CREATE TABLE products (
product_no integer UNIQUE,
name text,
price numeric );
```

5. Foreign Key Constarint

```
CREATE TABLE products (
product_no integer PRIMARY KEY,
name text,
price numeric );
```

CREATE TABLE orders (

order_id integer PRIMARY KEY,

product_no integer **REFERENCES products (product_no)**,

quantity integer

);

Here a foreign key constraint in the order table references the products table.

Modifying table:

Adding column

ALTER TABLE products ADD COLUMN description text;

Removing column

ALTER TABLE products DROP COLUMN description;

Adding Constraint

ALTER TABLE products ADD CONSTRAINT some_name UNIQUE (product_no);

ALTER TABLE products ADD FOREIGN KEY (product_group_id) REFERENCES product_groups;

Removing Constraint

ALTER TABLE products DROP CONSTRAINT some_name;

Adding Not Null Constraint

ALTER TABLE products ALTER COLUMN product_no SET NOT NULL;

Removing Not Null Constraint

ALTER TABLE products ALTER COLUMN product_no DROP NOT NULL;

Task1: Exercise –

1. Create table DEPT with the following columns and constraints

Column name	Data type	Size	Constraint
DEPTNO	NUMBER	2	PRIMARY KEY
DNAME	VARCHAR2	10	UNIQUE + NOT NULL
LOCATION	VARCHAR2	10	UNIQUE + NOT NULL

2. Create table EMPLOYEE with the following columns and constraints

Column name	Data type	Size	Constraint
EMPNO	CHAR	4	PRIMARY KEY
ENAME	VARCHAR2	10	NOT NULL
JOB	VARCHAR2	10	
MGR	CHAR	4	
HIREDATE	TIMESTAMP		NOT NULL
GENDER	CHAR	1	'M' OR 'F' ONLY
SAL	NUMBER	8,2	DEFAULT 0
COMM	NUMBER	8,2	DEFAULT 0
DEPTNO	NUMBER	2	FOREIGN KEY REFERRING TO DEPTNO of DEPT table

3. Insert 5 records in both the tables.

4. Add table level constraint such that commission cannot be greater than 30% of salary after the table has been created. Assign the constraint name COMM_30_SAL.

5. Add new constraint with the name DEPT_CHK_LOCATION to DEPT table such that LOCATION can be any one of the following cities MUMBAI, PUNE, BENGALURU, LONDON, SAN FRANSISCO only.

6. Remove the UNIQUE constraint from the LOCATION column.

Conclusion:

Thus using the schema diagram from the previous experiment, the tables were created using CREATE DDL statement with the primary key and foreign key constraint. Other constraints like Check, Unique and Not Null were added to the appropriate column by using ALTER DDL statement.

Post Lab Assignment:

1)What is NOT NULL constraint? What is DEFAULT constraint?
2) What is primary key? What is PRIMARY KEY constraint?
3) What is foreign key? How do you define a foreign key in your table?

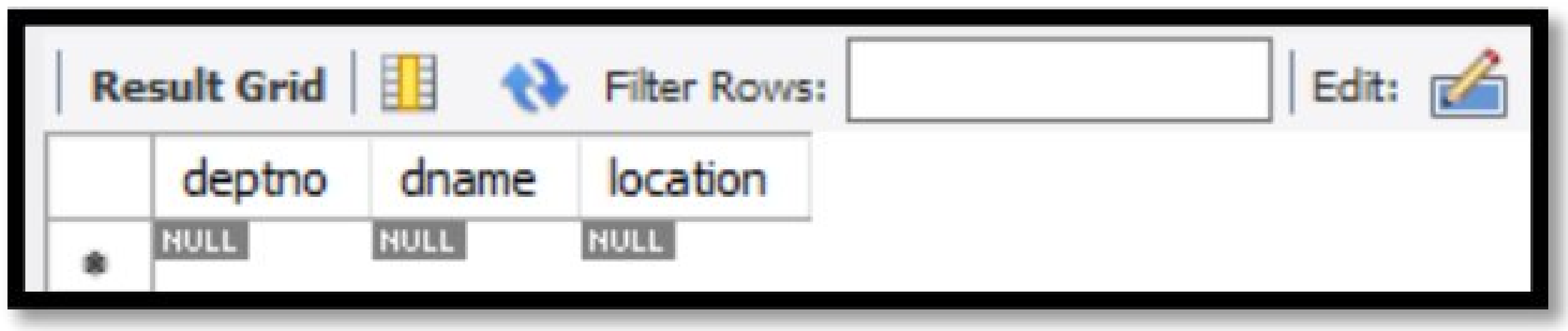
Signature of Faculty	Date of Completion:

1. **Create table DEPT with the following columns and constraints**

Query:

```sql
create table dept(
deptno numeric(2), PRIMARY KEY(deptno),
dname varchar(10) unique not null,
location varchar(10) unique not null);
```

Screenshot:



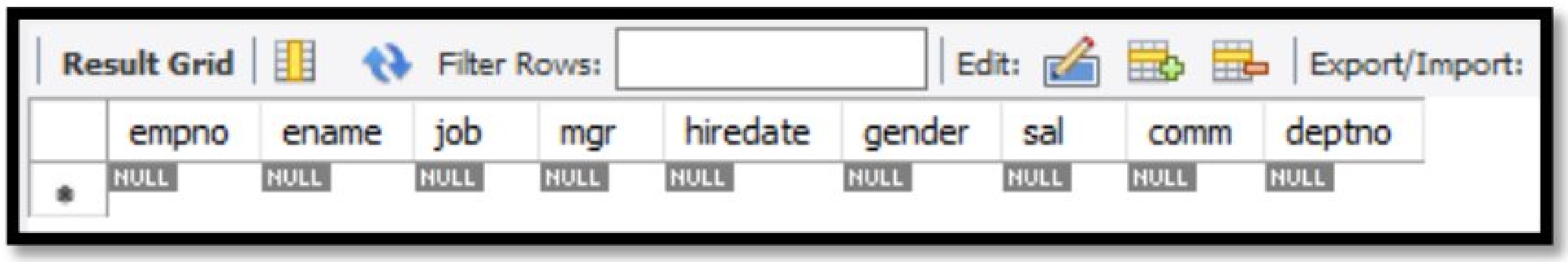
	deptno	dname	location
*	NULL	NULL	NULL

2. **Create table employee with the following columns and constraints**

Query:

```sql
create table employee(
empno char(4), PRIMARY KEY(empno),
ename varchar(10) not null,
job varchar(10),
mge char(4),
hiredate timestamp not null,
gender char(1), check(gender in ('M','F')),
sal numeric(8,2) default 0,
comm numeric(8,2) default 0,
deptno numeric(2),
constraint fk_deptno foreign key (deptno) references dept(deptno));
```

Screenshot:



	empno	ename	job	mgr	hiredate	gender	sal	comm	deptno
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

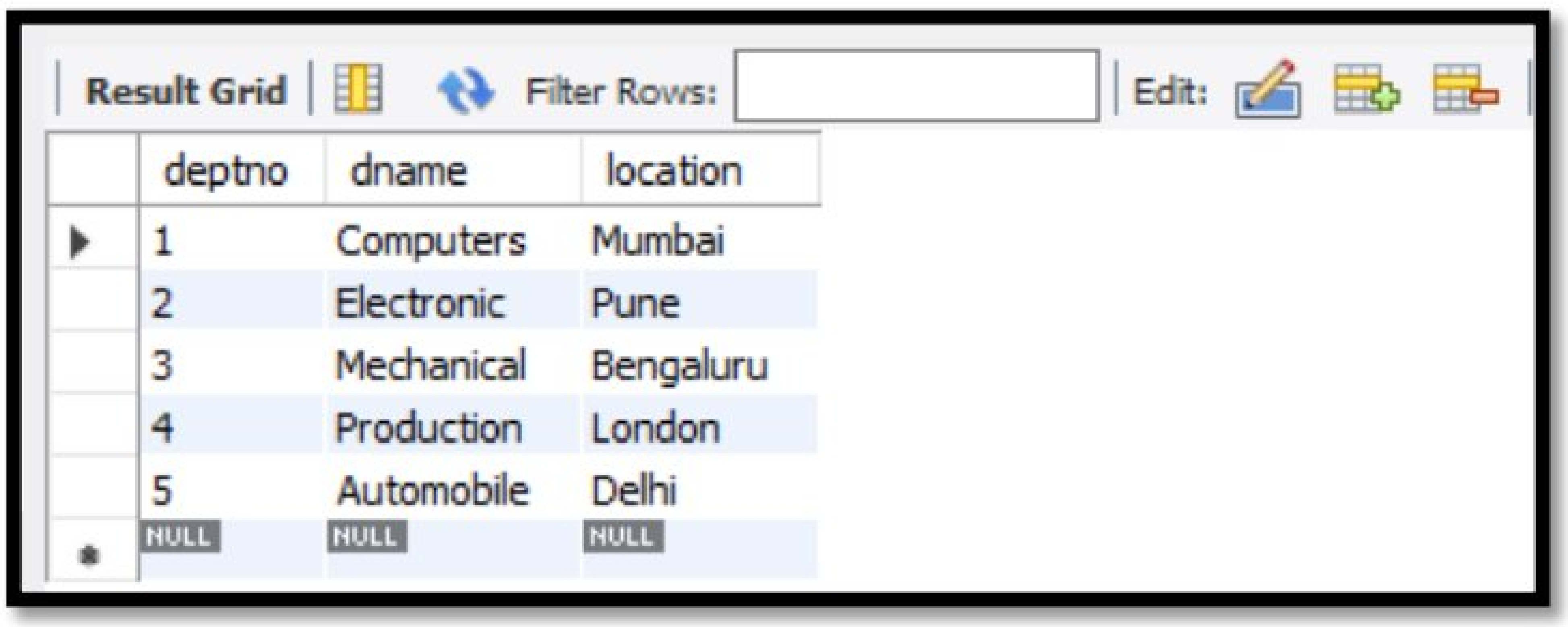
3. Insert 5 records in both the table.

A.] Dept table

Query:

```
Insert into dept value('001', 'Computers','Mumbai');
Insert into dept value('002', 'Electronic','Pune');
Insert into dept value('003', 'Mechanical','Bengaluru');
Insert into dept value('004', 'Production','London');
Insert into dept value('005', 'Automobile','Delhi');
```

Screenshot:



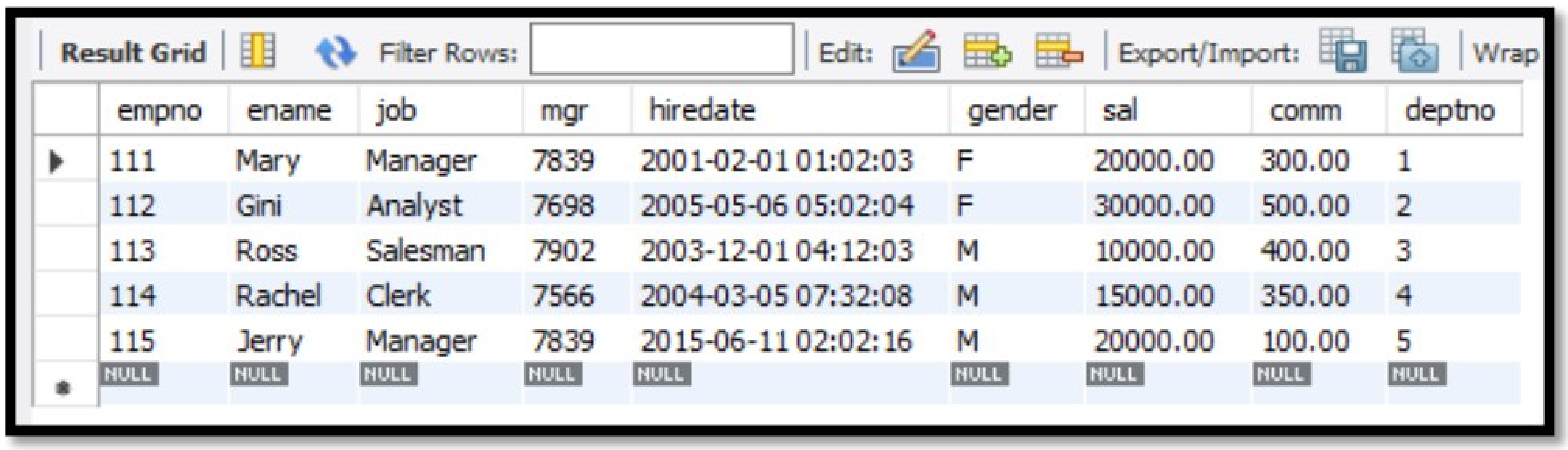
deptno	dname	location
1	Computers	Mumbai
2	Electronic	Pune
3	Mechanical	Bengaluru
4	Production	London
5	Automobile	Delhi
NULL	NULL	NULL

B.] Employee table

Query:

```
Insert into employee value('111','Mary','Manager','7839', '2001-02-01
01:02:03','F','20000','300','001');
Insert into employee value('112','Gini','Analyst','7698', '2005-05-06
05:02:04','F','30000','500','002');
Insert into employee value('113','Ross','Salesman','7902', '2003-12-01
04:12:03','M','10000','400','003');
Insert into employee value('114','Rachel','Clerk','7566', '2004-03-05
07:32:08','M','15000','350','004');
Insert into employee value('115','Jerry','Manager','7839', '2015-06-11
02:02:16','M','20000','100','005');
```

Screenshot:



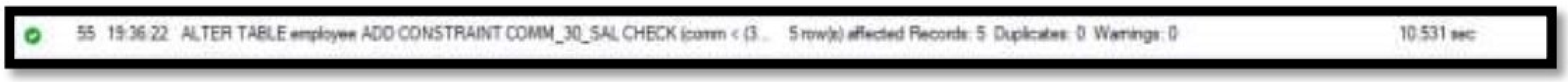
	empno	ename	job	mgr	hiredate	gender	sal	comm	deptno
▶	111	Mary	Manager	7839	2001-02-01 01:02:03	F	20000.00	300.00	1
	112	Gini	Analyst	7698	2005-05-06 05:02:04	F	30000.00	500.00	2
	113	Ross	Salesman	7902	2003-12-01 04:12:03	M	10000.00	400.00	3
	114	Rachel	Clerk	7566	2004-03-05 07:32:08	M	15000.00	350.00	4
	115	Jerry	Manager	7839	2015-06-11 02:02:16	M	20000.00	100.00	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. **Add table level constraint such that commission cannot be greater than 30% of salary after the table has been created. Assign the constraint name COMM_30_SAL.**

Query:

ALTER TABLE employee
ADD CONSTRAINT COMM_30_SAL CHECK (comm < (30/100) * sal);

Screenshot:



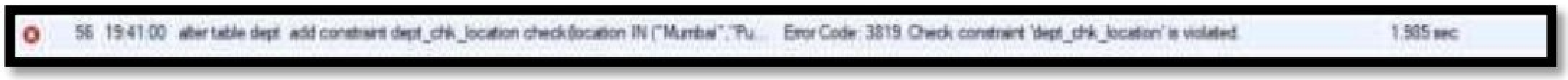
```
55  19:36:22  ALTER TABLE employee ADD CONSTRAINT COMM_30_SAL CHECK (comm < (3...   5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0      10.531 sec
```

5. **Add new constraint with the name DEPT_CHK_LOCATION to DEPT table such that LOCATION can be any one of the following cities MUMBAI, PUNE, BENGALURU, LONDON, SAN FRANSISCO only.**

Query:

alter table dept
add constraint dept_chk_location check(location IN
("Mumbai","Pune","Bengaluru","San Fransisco"));

Screenshot:



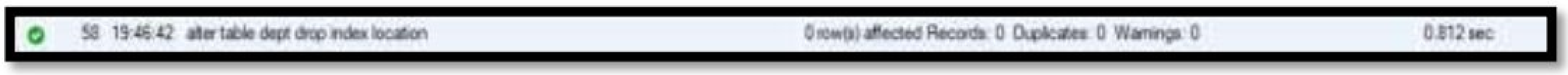
```
56  19:41:00  alter table dept  add constraint dept_chk_location check(location IN ("Mumbai","Pu...   Error Code: 3819 Check constraint 'dept_chk_location' is violated.      1.985 sec
```

6. Remove the UNIQUE constraint from the LOCATION column.

Query:

```
alter table dept
drop index location;
```

Screenshot:



```
58  19:46:42  alter table dept drop index location        0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0        0.812 sec
```

<u>Post Lab Assignment:</u>

1) What is NOT NULL constraint? What is DEFAULT constraint?

Ans:

<u>NOT NULL Constraint :</u>

The NOT NULL constraint is a column constraint that ensures values stored in a column are NOT NULL.
The syntax of defining a NOT NULL constraint is as follows:

column_name data_type NOT NULL;

A column may contain only one NOT NULL constraint which specifies a rule that the column must not contain any NULL value. In other words, if you update or insert NULL into a NOT NULL column, MySQL will issue an error.

DEFAULT Constraint:

The DEFAULT constraint is used to set a default value for a column.

The default value will be added to all new records, if no other value is specified.

The syntax of defining a DEFAULT constraint is as follows:

column_name data_type DEFAULT default_value;

2) What is primary key? What is PRIMARY KEY constraint?

Ans:

A primary key is a column or a set of columns that uniquely identifies each row in the table. The primary key follows these rules:

- A primary key must contain unique values. If the primary key consists of multiple columns, the combination of values in these columns must be unique.
- A primary key column cannot have NULL values. Any attempt to insert or update NULL to primary key columns will result in an error. Note that MySQL implicitly adds a NOT NULL constraint to primary key columns.
- A table can have one and only one primary key.

Because MySQL works faster with integers, the data type of the primary key column should be the integer e.g., INT, BIGINT. And you should ensure sure that value ranges of the integer type for the primary key are sufficient for storing all possible rows that the table may have.

A primary key column often has the AUTO_INCREMENT attribute that automatically generates a sequential integer whenever you insert a new row into the table.
When you define a primary key for a table, MySQL automatically creates an index called PRIMARY.

The PRIMARY KEY constraint allows you to define a primary key of a table when you create or alter table.

Typically, you define the primary key for a table in the CREATE TABLE statement. If the primary key has one column, you can use the PRIMARY KEY constraint as a column constraint:

```
CREATE TABLE table_name(
   primary_key_column datatype PRIMARY KEY,
   ...
);
```

When the primary key has more than one column, you must use the PRIMARY KEY constraint as a table constraint.

3) What is foreign key? How do you define a foreign key in your table?

Ans:

A foreign key is a column or group of columns in a table that links to a column or group of columns in another table. The foreign key places constraints on data in the related tables, which allows MySQL to maintain referential integrity.

Here is the basic syntax of defining a foreign key constraint in the CREATE TABLE or ALTER TABLE statement:

[CONSTRAINT constraint_name]
FOREIGN KEY [foreign_key_name] (column_name, ...)
REFERENCES parent_table(colunm_name,...)
[ON DELETE reference_option]
[ON UPDATE reference_option]

In this syntax:

First, specify the name of foreign key constraint that you want to create after the CONSTRAINT keyword. If you omit the constraint name, MySQL automatically generates a name for the foreign key constraint.
Second, specify a list of comma-separated foreign key columns after the FOREIGN KEY keywords. The foreign key name is also optional and is generated automatically if you skip it.
Third, specify the parent table followed by a list of comma-separated columns to which the foreign key columns reference.

Finally, specify how foreign key maintains the referential integrity between the child and parent tables by using the ON DELETE and ON UPDATE clauses. The reference_option determines action which MySQL will take when values in the parent key columns are deleted (ON DELETE) or updated (ON UPDATE).
MySQL has five reference options: CASCADE, SET NULL, NO ACTION, RESTRICT, and SET DEFAULT.