

# CASE STUDY

*Report submitted to*

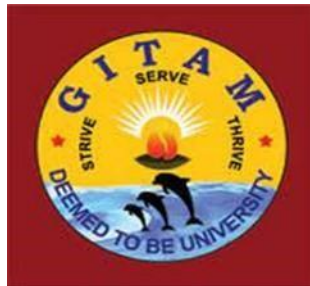
*Department of Computer Science and Engineering*

*By*

Registration ID	Name	Department
121910312048	N D Surya Prakash	CSE
121910312021	R Charan	CSE
121910312020	G Jeevan Reddy	CSE
121910312035	Pasyanthi Sagi	CSE

*Under The Guidance of*

***Prof K.THAMMI REDDY***



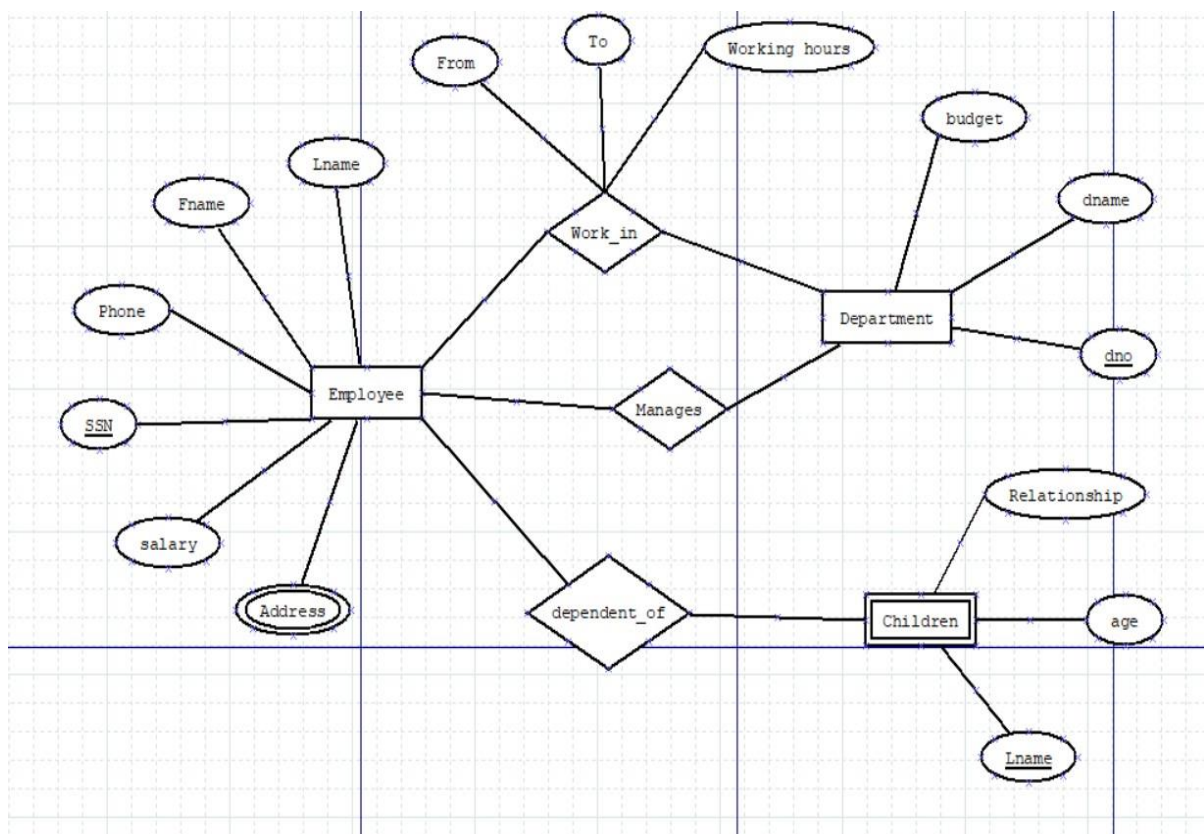
***DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING***

***GITAM Institute of Technology***

**AIM:** To Create a Company Database

**Description:** A company database needs to store information about employees (identified by ssn, with salary and phone as attributes), departments (identified by dno, with dname and budget as attributes), and children of employees (with name and age as attributes). Employees work in departments; each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known. (Assume if any data is required)

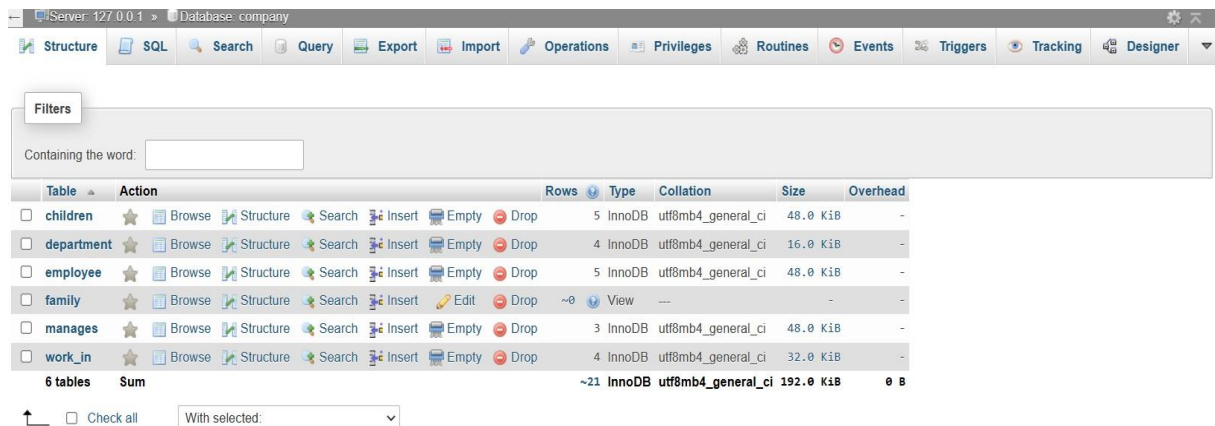
## **ER Diagram for Company Data base**



# MySQL using Xampp

## To Create Database :

CREATE DATABASE company;



The screenshot shows the MySQL Workbench interface for a database named 'company'. The 'Structure' tab is active, displaying a list of tables. Each table row includes a checkbox, a star icon, a 'Browse' button, and icons for 'Structure', 'Search', 'Insert', 'Empty', and 'Drop'. The table details include the number of rows, storage engine (InnoDB), collation (utf8mb4\_general\_ci), size, and overhead.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> children	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> department	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> employee	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> family	Browse Structure Search Insert Edit Drop View	~0	VIEW	---	-	-
<input type="checkbox"/> manages	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> work_in	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<b>6 tables</b>	<b>Sum</b>	<b>~21</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>192.0 KiB</b>	<b>0 B</b>

At the bottom, there is a 'Filters' section with a search box and a 'Check all' checkbox.

## Relational Database Schema:

### To Create Table Employee :

```
CREATE TABLE employee(  
    SSN int PRIMARY KEY,  
    Fname varchar(15) NOT null UNIQUE,  
    Lname varchar(15) NOT null,  
    Address varchar(255) not null,  
    phone char(10),  
    salary decimal(10,2)  
);
```

To Add into Table Employee :

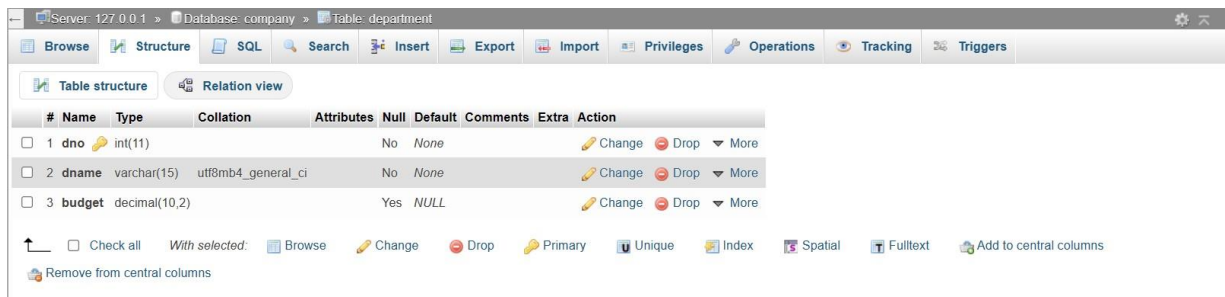
```
ALTER TABLE employee add Dnumber int;
```

```
Alter table employee add foreign key (Dnumber) references Department (Dno);
```



## To Create Table department :

```
CREATE TABLE department(  
    dno int PRIMARY KEY,  
    dname varchar(15) NOT null,  
    budget decimal(10,2));
```



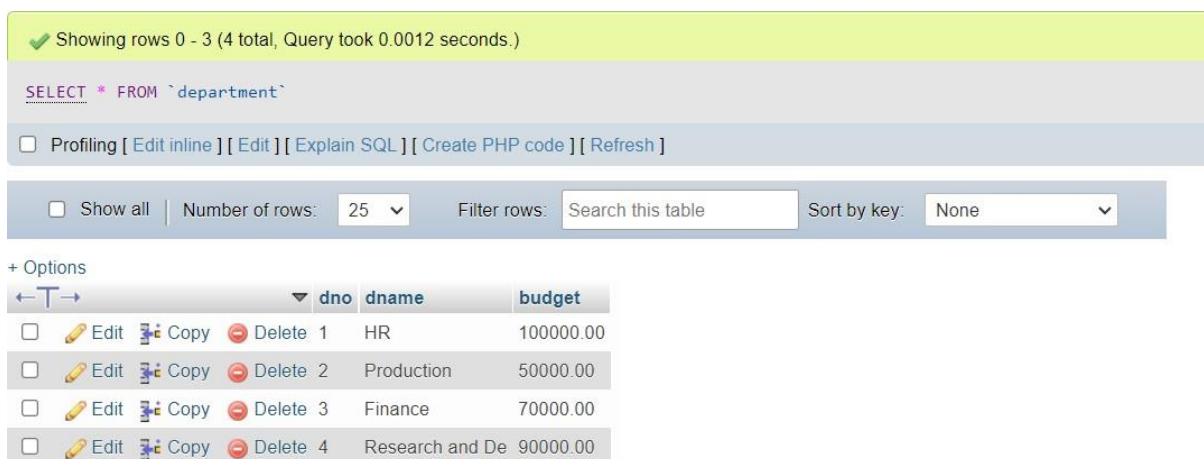
The screenshot shows the phpMyAdmin interface for a database named 'company'. The 'Table: department' tab is selected. The 'Table structure' view is active, displaying the following table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	dno	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	dname	varchar(15)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	budget	decimal(10,2)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the table structure, there are options to 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', and 'Add to central columns'. There is also a 'Remove from central columns' option.

## Insert Values into Department Table:

```
INSERT INTO `department` (`dno`, `dname`, `budget`) VALUES  
(1, 'HR', '100000.00'),  
(2, 'Production', '50000.00'),  
(3, 'Finance', '70000.00')  
(4, 'Research and De', '90000.00')
```



The screenshot shows the phpMyAdmin interface displaying the results of an INSERT query. The status bar indicates 'Showing rows 0 - 3 (4 total, Query took 0.0012 seconds.)'. The SQL query is: `SELECT * FROM `department``. Below the query, there are options for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. The table structure is shown with columns: dno, dname, and budget. The data is displayed in a table with 4 rows:

	dno	dname	budget
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	HR	100000.00
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	Production	50000.00
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3	Finance	70000.00
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4	Research and De	90000.00

## To Create Table children:

```
CREATE TABLE children(  
    ESSN int,  
    Fname varchar(15),  
    Lname varchar(15),  
    age int(2),  
    relationship varchar(10),  
    PRIMARY KEY(ESSN, Lname),  
    FOREIGN KEY (ESSN) REFERENCES employee(SSN) ON DELETE CASCADE,  
    FOREIGN KEY (Fname) REFERENCES employee (Fname) ON DELETE CASCADE  
);
```



The screenshot shows a database management interface with a toolbar at the top containing buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers. Below the toolbar, the 'Table structure' tab is selected for the 'children' table. The table structure is displayed in a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The table has five columns: ESSN (int(11)), Fname (varchar(15)), Lname (varchar(15)), age (int(2)), and relationship (varchar(10)). All columns are nullable and have a default value of NULL. Each row has a 'Change' button, a 'Drop' button, and a 'More' dropdown menu.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ESSN	int(11)			Yes	NULL			Change Drop More
2	Fname	varchar(15)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	Lname	varchar(15)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	age	int(2)			Yes	NULL			Change Drop More
5	relationship	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More

## Insert Values into Children Table:

```
INSERT INTO `children` (`ESSN`, `Fname`, `Lname`, `age`, `relationship`) VALUES  
(10001, 'Nagamani', 'Karthik', 12, 'Son'),  
(10001, 'Nagamani', 'Surya', 18, 'Son'),  
(10002, 'Valasali', 'Sai', 11, 'Son'),  
(10003, 'Rekala', 'Jaanu', 8, 'Daughter'),  
(10003, 'Rekala', 'Surya', 12, 'Son');
```

Server: 127.0.0.1 » Database: company » Table: children

⚙️ Browse ⚙️ Structure ⚙️ SQL ⚙️ Search ⚙️ Insert ⚙️ Export ⚙️ Import ⚙️ Privileges ⚙️ Operations ⚙️ Tracking ⚙️ Triggers

⚠️ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 4 (5 total, Query took 0.0014 seconds.)

`SELECT * FROM `children``

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

ESSN	Fname	Lname	age	relationship
10001	Nagamani	Karthik	12	Son
10001	Nagamani	Surya	18	Son
10002	Valasali	Sai	11	Son
10003	Rekala	Jaanu	8	Daughter
10003	Rekala	Surya	12	Son

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

## To Create Table work\_in:

CREATE TABLE work\_in(

ESSN int PRIMARY KEY,

Dnumber int,

From\_ date;

To\_ date;

Working\_hrs time,

FOREIGN KEY (ESSN) REFERENCES employee(SSN) ON DELETE CASCADE,

FOREIGN KEY (Dnumber) REFERENCES department(dno) ON DELETE CASCADE ON UPDATE CASCADE);

Server: 127.0.0.1 » Database: company » Table: work\_in

⚙️ Browse ⚙️ Structure ⚙️ SQL ⚙️ Search ⚙️ Insert ⚙️ Export ⚙️ Import ⚙️ Privileges ⚙️ Operations ⚙️ Tracking ⚙️ Triggers

⚙️ Table structure ⚙️ Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	ESSN	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	Dnumber	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	From_	date			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	To_	date			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	working_hrs	time			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>



## Insert Values into Work\_in Table:

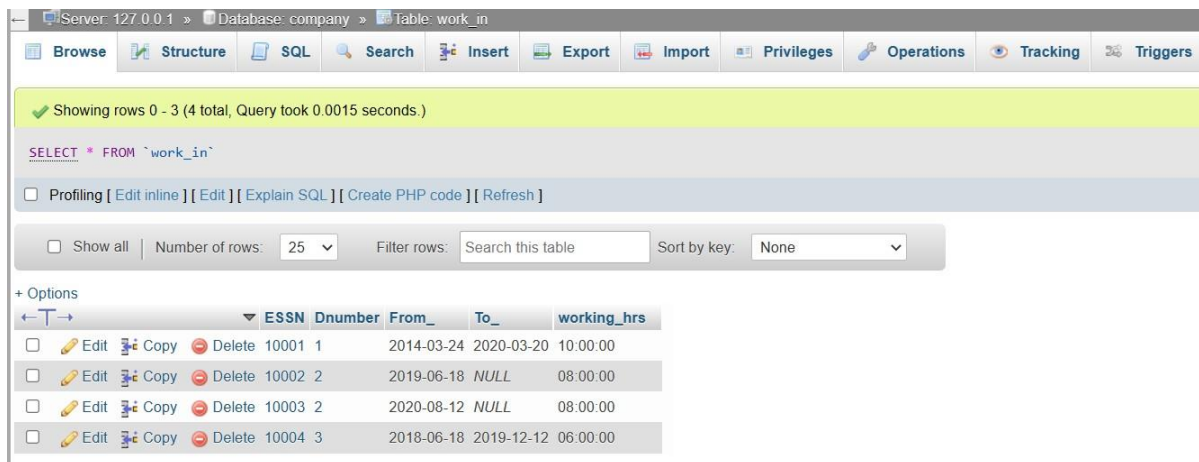
```
INSERT INTO `work_in` (`ESSN`, `Dnumber`, `From_`, `To_`, `working_hrs`)
VALUES
```

```
(10001, 1, '2014-03-24', '2020-03-20', '10:00:00'),
```

```
(10002, 2, '2019-06-18', NULL, '08:00:00'),
```

```
(10003, 2, '2020-08-12', NULL, '08:00:00'),
```

```
(10004, 3, '2018-06-18', '2019-12-12', '06:00:00');
```



The screenshot shows a database management interface for a table named 'work\_in'. The table has 4 rows of data. The columns are ESSN, Dnumber, From\_, To\_, and working\_hrs. The data is as follows:

	ESSN	Dnumber	From_	To_	working_hrs
<input type="checkbox"/>	10001	1	2014-03-24	2020-03-20	10:00:00
<input type="checkbox"/>	10002	2	2019-06-18	NULL	08:00:00
<input type="checkbox"/>	10003	2	2020-08-12	NULL	08:00:00
<input type="checkbox"/>	10004	3	2018-06-18	2019-12-12	06:00:00

## To Create Table manages:

```
CREATE TABLE manages(
```

```
    ESSN int UNIQUE NOT null,
```

```
    Dnumber int PRIMARY KEY,
```

```
    FOREIGN KEY (ESSN) REFERENCES employee (SSN) ON DELETE
    CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (Dnumber) REFERENCES department (dno));
```



The screenshot shows the 'Table structure' view for a table named 'manages'. The table has 2 columns: Dnumber and ESSN. Both columns are of type int(11) and have no default values or comments.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Dnumber	int(11)			No	None			Change Drop More
2	ESSN	int(11)			No	None			Change Drop More



## Insert Values into manages Table:

```
INSERT INTO `manages` (`Dnumber`, `ESSN`) VALUES
```

```
(1, 10003),
```

```
(2, 10002),
```

```
(3, 10004);
```



Server: 127.0.0.1 » Database: company » Table: manages

Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

`SELECT * FROM `manages``

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

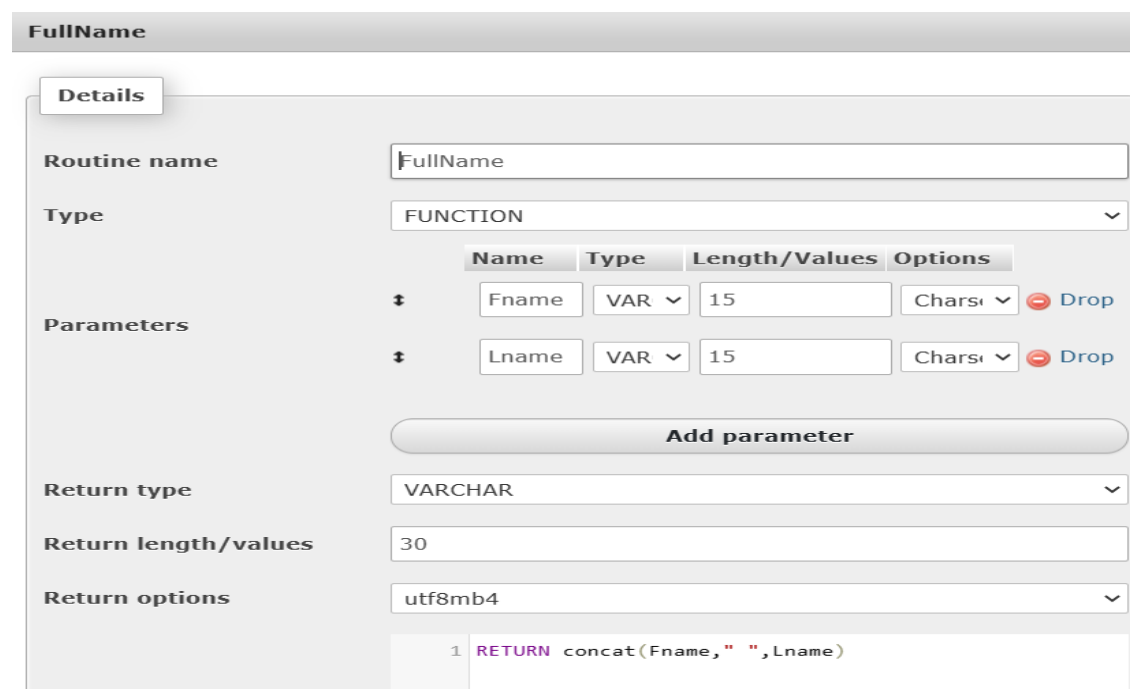
	Dnumber	ESSN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	10002
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	10003
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	10004

## To Create Function:

```
CREATE FUNCTION Name(Fname varchar(15),Lname varchar(15))
```

```
RETURNS varchar(30) DETERMINISTIC
```

```
RETURN concat(Fname,' ',Lname);
```



**FullName**

Details

Routine name: FullName

Type: FUNCTION

Parameters:

Name	Type	Length/Values	Options
Fname	VAR	15	Chars: Drop
Lname	VAR	15	Chars: Drop

Add parameter

Return type: VARCHAR

Return length/values: 30

Return options: utf8mb4

```
1 RETURN concat(Fname," ",Lname)
```

## To Create View:

CREATE VIEW family AS

```
SELECT FullName(e.Fname,e.Lname)
Parent_Name,FullName(c.Fname,c.Lname) child_Name

FROM children c,employee e

WHERE e.SSN=c.ESSN
```

Server: 127.0.0.1 » Database: company » View: family

Browse Structure SQL Search Export Privileges Operations Tracking

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 4 (5 total, Query took 0.1202 seconds.)

```
SELECT * FROM `family`
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Parent_Name	child_Name
Nagamani Prakash	Nagamani Karthik
Nagamani Prakash	Nagamani Surya
Valasali Mouli	Valasali Sai
Rekala Charan	Rekala Jaanu
Rekala Charan	Rekala Surya

## Data Connectivity:

### For Connection:

```
import mysql.connector as c

con = c.connect(host="localhost", user="root", passwd="", database="company")

if con.is_connected():

    print("sucessfully connected")

cursor = con.cursor()
```

## For Employee Table:

```
def employee():

    def insert():

        SSN = int(input("Enter Employee ID: "))

        Fname = input("First Name: ")

        Lname = input("Last Name: ")

        Address = input("Address: ")

        phone = input("phone: ")

        salary = int(input("Enter Salary: "))

        Dnumber = int(input("Enter Department No: "))

        query = "insert into employee values\n({},{},{},{},{},{},{})".format(SSN,Fname,Lname,Address,phone,salary,Dnumber)

        cursor.execute(query)

        con.commit()

        print("Data Inserted Sucessfully")


    def update():

        def update_salary():

            SSN = int(input("Enter Employee ID: "))

            salary = int(input("Enter Salary: "))

            query = "update employee set salary = {} where SSN={}".format(salary,SSN)

            cursor.execute(query)

            con.commit()

            if cursor.rowcount>0:

                print("salary of {} Updated Sucessfully".format(SSN))

            else:

                print("No Data Found")

        def update_phone():

            SSN = int(input("Enter Employee ID: "))

            phone = input("Enter New phone no.: ")

            query = "update employee set phone = '{}' where SSN={}".format(phone,SSN)

            cursor.execute(query)

            con.commit()
```

```

        if cursor.rowcount>0:

            print("Phone no. of {} Updated Successfully".format(SSN))

        else:

            print("No Data Found")

def update_Deptno():

    SSN = int(input("Enter Employee ID: "))

    Dnumber = int(input("Enter New Dept no.: "))

    query = "update employee set Dnumber = {} where SSN={}".format(Dnumber,SSN)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Dept no. of {} Updated Successfully".format(SSN))

    else:

        print("No Data Found")

def update_Address():

    SSN = int(input("Enter Employee ID: "))

    Address = input("Enter New Address: ")

    query = "update employee set Address = '{}' where SSN={}".format(Address,SSN)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Address of {} Updated Successfully".format(SSN))

    else:

        print("No Data Found")

print("Updating Employee Details.")

while True:

    n = int(input("1:Update Salary 2:Update Phone no. 3:Dept no. 4:Address 5:Exit\nEnter
Choice: "))

    if n == 1:

        update_salary()

    elif n ==2:

        update_phone()

    elif n ==3:

```

```

        update_Deptno()

    elif n == 4:

        update_Address()

    elif n ==5:

        break

    else:

        print("Enter a valid number")

def delete():

    SSN = int(input("Enter Employee ID: "))

    query = "Delete from employee where SSN={}".format(SSN)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Deleted employee ID = {} Sucessfully".format(SSN))

    else:

        print("No Data Found")

while True:

    a = int(input("1:Insert 2:Update 3:Delete 4:Exit "))

    if a == 1:

        insert()

    elif a == 2:

        update()

    elif a == 3:

        delete()

    elif n ==4:

        break

    else:

        print("Enter valid number")

```

## For Department Table:

```
def department():

    def insert():

        dno = input("Department Number: ")

        dname = input("Department Name: ")

        budget = input("Budget: ")

        query = "insert into department values ({},'{}'.format(dno,dname,budget)

        cursor.execute(query)

        con.commit()

        print("Data Inserted Sucessfully")

    def delete():

        dno = int(input("Enter department Number: "))

        query = "Delete from department where dno={}".format(dno)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Deleted department number {} Sucessfully".format(dno))

        else:

            print("No Data Found")

    def update():

        def update_dept_name():

            dno = int(input("Enter department No. : "))

            dname = input("Enter New dept Name: ")

            query = "update department set dname = '{}' where dno={}".format(dname,dno)

            cursor.execute(query)

            con.commit()

            if cursor.rowcount>0:

                print("department Name of {} Updated Sucessfully".format(dno))

            else:

                print("No Data Found")

        def update_budget():

            dno = int(input("Enter department No. : "))
```

```

        budget = int(input("Enter New budget: "))

        query = "update department set budget = {} where dno={}".format(budget,dno)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("budget of {} Updated Sucessfully".format(dno))

        else:

            print("No Data Found")


    print("Updating department Details.")

    while True:

        n = int(input("1:Update department Name 2:Update budget 3:Exit\nEnter Choice: "))

        if n == 1:

            update_dept_name()

        elif n ==2:

            update_budget()

        elif n ==3:

            break

        else:

            print("Enter a valid number")

    while True:

        a = int(input("1:insert 2:Update 3:Delete 4:Exit "))

        if a == 1:

            insert()

        elif a == 2:

            update()

        elif a == 3:

            delete()

        elif a ==4:

            break

        else:

            print("Enter valid number")

```



## For Work in Table:

```
def work_in():

    def insert():

        ESSN = int(input("Enter Employee ID: "))

        Dnumber = input("Department Number: ")

        From_ = input("Date of Joining: ")

        To_ = input("Date of Leaving: ")

        working_hrs = input("No. Of Hrs/Day: ")

        query = "insert into work_in values\n({},{},'{}','{}','{}')".format(ESSN,Dnumber,From_,To_,working_hrs)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Data Inserted Sucessfully")

        else:

            print("No Data Found")

    def delete():

        ESSN = int(input("Enter Employee ID: "))

        query = "Delete from work_in where ESSN={}".format(ESSN)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Deleted employee ID = {} Sucessfully".format(ESSN))

        else:

            print("No Data Found")

    def update():

        def update_dept_no():

            ESSN = int(input("Enter Employee ID: "))

            Dnumber = int(input("Enter New department No. : "))

            query = "update work_in set Dnumber = {} where ESSN={}".format(Dnumber,ESSN)

            cursor.execute(query)

            con.commit()

            if cursor.rowcount>0:
```

```

        print("department Number of {} Updated Sucessfully".format(ESSN))

    else:

        print("No Data Found")

def update_dateOfjoining():

    ESSN = int(input("Enter Employee ID: "))

    From_ = input("Enter Date of Joining : ")

    query = "update work_in set From_ = '{}' where ESSN={}".format(From_,ESSN)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Date of Joining {} Updated Sucessfully".format(ESSN))

    else:

        print("No Data Found")

def update_dateOfleaving():

    ESSN = int(input("Enter Employee ID: "))

    To_ = input("Enter Date of Leaving : ")

    query = "update work_in set To_ = '{}' where ESSN={}".format(To_,ESSN)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Date of Leaving {} Updated Sucessfully".format(ESSN))

    else:

        print("No Data Found")

print("Updating Work Details.")

while True:

    n = int(input("1:Update department Name 2:Update dateOfjoining 3: update dateOfleaving
4:Exit\nEnter Choice: "))

    if n == 1:

        update_dept_no()

    elif n ==2:

        update_dateOfjoining()

    elif n ==3:

```

```

        update_dateOfleaving()

    elif n ==4:

        break

    else:

        print("Enter a valid number")

while True:

    a = int(input("1:insert 2:Update 3:Delete 4:Exit "))

    if a == 1:

        insert()

    elif a == 2:

        update()

    elif a == 3:

        delete()

    elif a ==4:

        break

    else:

        print("Enter valid number")

```

## For manages Table:

```

def manages():

    def insert():

        ESSN = int(input("Enter Employee ID: "))

        Dnumber = input("Department Number: ")

        query = "insert into manages values ({},{})".format(Dnumber,ESSN)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Data Inserted Sucessfully")

        else:

            print("No Data Found")

    def delete():

        ESSN = int(input("Enter Employee ID to Delete: "))

```

```

query = "delete from manages where ESSN={}".format(ESSN)

cursor.execute(query)

con.commit()

if cursor.rowcount>0:

    print("Data Deleted Sucessfully")

else:

    print("No Data Found")

def update():

    Dnumber = int(input("Enter department No. : "))

    ESSN = int(input("Enter New Manager ID: "))

    query = "update manages set ESSN = {} where Dnumber={}".format(ESSN,Dnumber)

    cursor.execute(query)

    con.commit()

    if cursor.rowcount>0:

        print("Manager of Department {} Updated Sucessfully".format(Dnumber))

    else:

        print("No Data Found")

while True:

    a = int(input("1:insert 2:Update 3:Delete 4:Exit "))

    if a == 1:

        insert()

    elif a == 2:

        update()

    elif a == 3:

        delete()

    elif a ==4:

        break

    else:

        print("Enter valid number")

```

## For Children Table:

```
def children():

    def insert():

        ESSN = int(input("Enter Employee ID: "))

        Fname = input("Enter First Name: ")

        Lname = input("Enter Last Name: ")

        age = int(input("Enter age: "))

        relationship = input("Enter son/daughter/other: ")

        query = "insert into children values\n({},{},{},{})".format(ESSN,Fname,Lname,age,relationship)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Data Inserted Sucessfully")

        else:

            print("No Data Found")

    def delete():

        ESSN = int(input("Enter Employee ID to Delete: "))

        Lname = input("Enter Child Name: ")

        query = "delete from children where ESSN={} and Lname='{}'".format(ESSN,Lname)

        cursor.execute(query)

        con.commit()

        if cursor.rowcount>0:

            print("Data Deleted Sucessfully")

        else:

            print("No Data Found")

    def update():

        ESSN = int(input("Enter Employee ID: "))

        Lname = input("Enter child Name: ")

        age = int(input("Enter current age: "))

        query = "update children set age = {} where ESSN={} and Lname='{}'".format(age,ESSN,Lname)

        cursor.execute(query)
```

```

con.commit()

if cursor.rowcount>0:

    print("Manager of Department {} Updated Sucessfully".format(ESSN))

else:

    print("No Data Found")

while True:

    a = int(input("1:insert 2:Update 3:Delete 4:Exit "))

    if a == 1:

        insert()

    elif a == 2:

        update()

    elif a == 3:

        delete()

    elif a ==4:

        break

    else:

        print("Enter valid number")

```

## Main Function:

```

if __name__ == '__main__':

    print("Enter choice: ")

    while True:

        n = int(input("1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit "))

        if n == 1:

            employee()

        elif n ==2:

            department()

        elif n ==3:

            work_in()

        elif n == 4:

            manages()

```

```
elif n == 5:

    children()

elif n == 6:

    break
```

## Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\company> conda activate base
PS C:\xampp\htdocs\company> & C:/Users/praka/anaconda3/python.exe c:/xampp/htdocs/company/main.py
sucessfully connected
Enter choice:
1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit █
```

## Example:

Enter any Choice from 1 to 5:

```
Enter choice:
1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit 1
1:insert 2:Update 3:Delete 4:Exit █
```

Now, You can Insert/Update/Delete any Tables You want.

If we Choose **1: Insert From Employee Table**, We need to Enter Following Details.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit 1
1:insert 2:Update 3:Delete 4:Exit 1
Enter Employee ID: 111
First Name: prakash
Last Name: nagamani
Address: mvp colony
phone: 986856552
Enter Salary: 100000
Enter Department No: 2
Data Inserted Sucessfully
1:insert 2:Update 3:Delete 4:Exit █
```



## Output:

+ Options									
← T →									
SSN Fname Lname Address phone salary Dnumber									
<input type="checkbox"/>	Edit	Copy	Delete	111	prakash	nagamani	mvp colony	986856552	100000.00 2

If we Choose **2: Update From Employee Table**, We need to Enter Following Details.

We can Update Following Details:

1:Update Salary 2:Update Phone no. 3:Dept no. 4:Address 5:Exit

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\prakash\SEM 5\DBMS\CASESTUDY> conda activate base
PS D:\prakash\SEM 5\DBMS\CASESTUDY> & C:/Users/praka/anaconda3/python.exe "d:/prakash/SEM 5/DBMS/CASESTUDY/main.py"
sucessfully connected
Enter choice:
1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit 1
1:insert 2:Update 3>Delete 4:Exit 2
Updating Employee Details.
1:Update Salary 2:Update Phone no. 3:Dept no. 4:Address 5:Exit
Enter Choice: 1
Enter Employee ID: 111
Enter Salary: 500000
salary of 111 Updated Sucessfully
1:Update Salary 2:Update Phone no. 3:Dept no. 4:Address 5:Exit
Enter Choice: 5
1:insert 2:Update 3>Delete 4:Exit
```

## Output:

+ Options									
← T →									
SSN Fname Lname Address phone salary Dnumber									
<input type="checkbox"/>	Edit	Copy	Delete	111	prakash	nagamani	mvp colony	986856552	500000.00 2

If we Choose **3: Delete From Children Table**, We need to Enter Following Details.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\prakash\SEM 5\DBMS\CASESTUDY> conda activate base
PS D:\prakash\SEM 5\DBMS\CASESTUDY> & C:/Users/praka/anaconda3/python.exe "d:/prakash/SEM 5/DBMS/CASESTUDY/main.py"
sucessfully connected
Enter choice:
1:Employee 2:Department 3:Work In 4:Manages 5:Children 6:Exit 5
1:insert 2:Update 3>Delete 4:Exit 3
Enter Employee ID to Delete: 10002
Enter Child Name: sai
Data Deleted Sucessfully
1:insert 2:Update 3>Delete 4:Exit
```