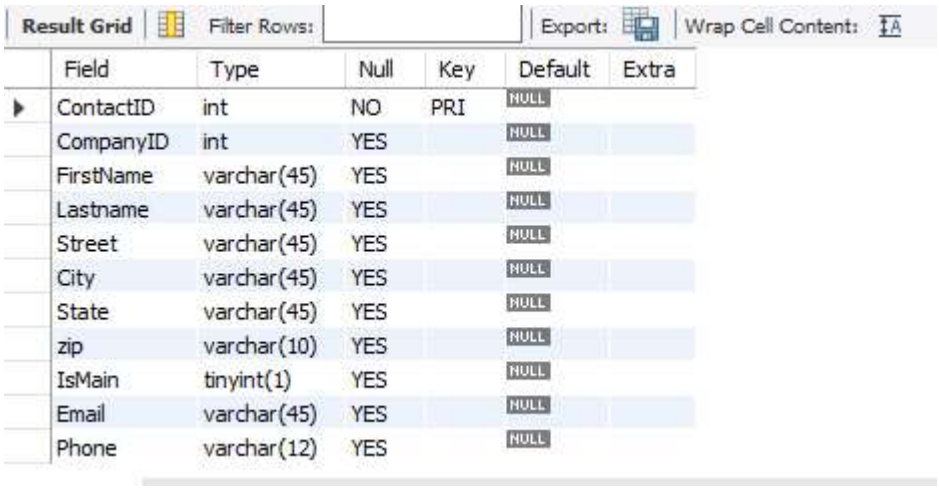
Q- 1: Statement to create the Contact table

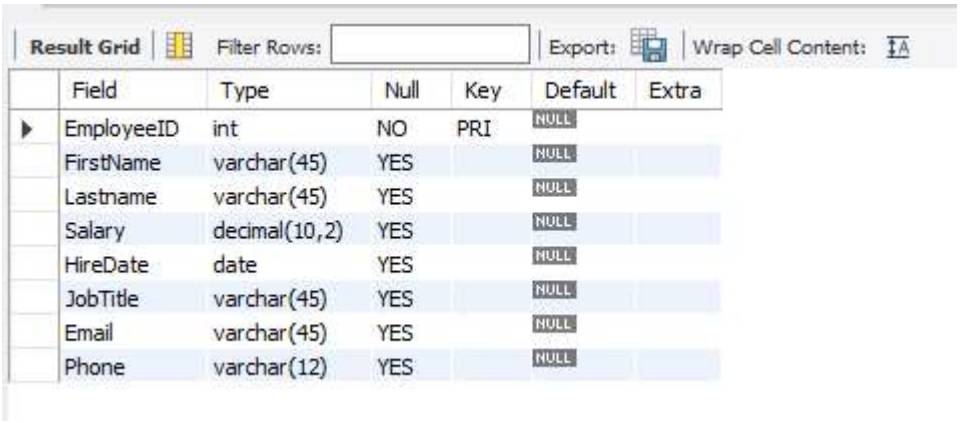
- create table Contact(
  ContactID int primary key ,
  CompanyID int,
  FirstName varchar(45),
  Lastname varchar(45),
  Street varchar(45),
  City varchar(45),
  State varchar(45),
  zip varchar(10),
  IsMain boolean,
  Email varchar(45),
  Phone varchar(12)
  );



Field	Type	Null	Key	Default	Extra
ContactID	int	NO	PRI	NULL	
CompanyID	int	YES		NULL	
FirstName	varchar(45)	YES		NULL	
Lastname	varchar(45)	YES		NULL	
Street	varchar(45)	YES		NULL	
City	varchar(45)	YES		NULL	
State	varchar(45)	YES		NULL	
zip	varchar(10)	YES		NULL	
IsMain	tinyint(1)	YES		NULL	
Email	varchar(45)	YES		NULL	
Phone	varchar(12)	YES		NULL	

Q-2 Statement to create the Employee table

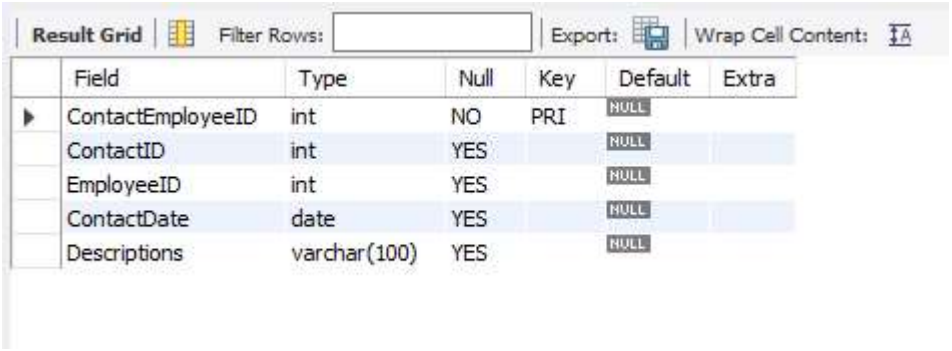
- create table Employee(
  EmployeeID int primary key,
  FirstName varchar(45),
  Lastname varchar(45),
  Salary decimal(10,2),
  HireDate date,
  JobTitle varchar(45),
  Email varchar(45),
  Phone varchar(12)
  );



Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	
FirstName	varchar(45)	YES		NULL	
Lastname	varchar(45)	YES		NULL	
Salary	decimal(10,2)	YES		NULL	
HireDate	date	YES		NULL	
JobTitle	varchar(45)	YES		NULL	
Email	varchar(45)	YES		NULL	
Phone	varchar(12)	YES		NULL	

Q-3 Statement to create the ContactEmployee table

- create table ContactEmployee(
  ContactEmployeeID int primary key,
  ContactID int,
  EmployeeID int,
  ContactDate Date,
  Descriptions varchar(100)
  );

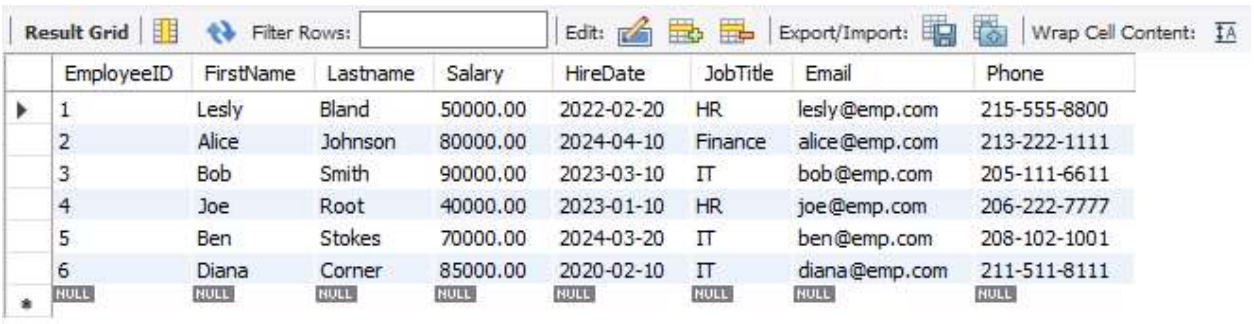


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
ContactEmployeeID	int	NO	PRI	NULL	
ContactID	int	YES		NULL	
EmployeeID	int	YES		NULL	
ContactDate	date	YES		NULL	
Descriptions	varchar(100)	YES		NULL	

Q -4 In the Employee table, the statement that changes Lesley Bland's phone number to 215-555-8800

- update Employee
  set phone = "215-555-8800"
  where FirstName = "Lesly" and Lastname = "Bland";

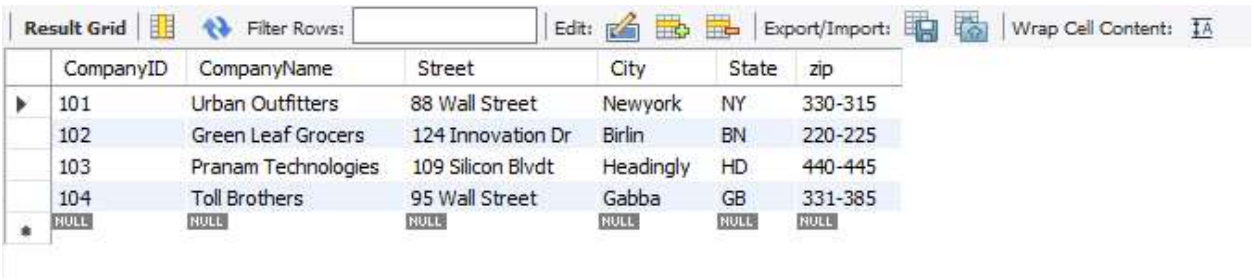


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmployeeID	FirstName	Lastname	Salary	HireDate	JobTitle	Email	Phone
1	Lesly	Bland	50000.00	2022-02-20	HR	lesly@emp.com	215-555-8800
2	Alice	Johnson	80000.00	2024-04-10	Finance	alice@emp.com	213-222-1111
3	Bob	Smith	90000.00	2023-03-10	IT	bob@emp.com	205-111-6611
4	Joe	Root	40000.00	2023-01-10	HR	joe@emp.com	206-222-7777
5	Ben	Stokes	70000.00	2024-03-20	IT	ben@emp.com	208-102-1001
6	Diana	Corner	85000.00	2020-02-10	IT	diana@emp.com	211-511-8111
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Q-5 In the Company table, the statement that changes the name of “Urban Outfitters, Inc.” to “Urban Outfitters” .

- update Company
  set CompanyName = "Urban Outfitters"
  where CompanyName = "Urban Outfitter's,Inc";



CompanyID	CompanyName	Street	City	State	zip
101	Urban Outfitters	88 Wall Street	Newyork	NY	330-315
102	Green Leaf Grocers	124 Innovation Dr	Birlin	BN	220-225
103	Pranam Technologies	109 Silicon Blvdt	Headingly	HD	440-445
104	Toll Brothers	95 Wall Street	Gabba	GB	331-385
NULL	NULL	NULL	NULL	NULL	NULL

Q-6 In ContactEmployee table, the statement that removes Dianne Connor's contact event with Jack Lee (one statement).

- ```
  with Emp_name as(
  select
  c.contactEmployeeId,c.ContactId,e.firstname,e.lastname
     from employee e
     join contactemployee c
     on e.employeeid = c.employeeid
     where Firstname = "Diana" and lastname = "Corner"
  )
  delete et
  from contact et
  join Emp_name en
  on et.contactid = en.contactid
  where en.firstname = "Diana" and et.firstname = "Jack";
  ```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ContactID	CompanyID	FirstName	Lastname	Street	City	State	zip	IsMain	Email	Phone
112	102	Aric	Johnson	124 Innovation Dr	Birlin	BN	220-225	1	aric@contact.com	314-303-7766
113	103	Bruce	Lee	109 Silicon Blvdt	Headingly	HD	440-445	1	bruce@contact.com	315-304-6677
114	104	Mitch	Marsh	95 Wall Street	Gabba	GB	331-385	1	mitch@contact.com	314-310-7766
115	104	Trent	Boult	95 Wall Street	Gabba	GB	331-385	1	trent@contact.com	315-311-6677
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
  ```

Q - 7 Write the SQL SELECT query that displays the names of the employees that have contacted Toll Brothers (one statement).

- select e.Firstname
  from Company co
  join contact cn
  on co.companyId = cn.companyid
  join contactemployee ce
  on cn.contactid = ce.contactid
  join employee e
  on ce.employeeid = e.employeeid
  where co.companyname = "Toll Brothers";

Module:4 Working with Database Using SQL

Q-8 What is the significance of "%" and "_" operators in the Like statement.

Sol^n ("%") - Percent Sign:-

- Represent zero, one or many characters.
- It is used when you what to match any number of characters (including none).

-> ("_") - Underscore

- Represents a single character only.
- Use it when you want to match exactly one character in position.

Q-9 Explain normalization in the context of databases.

Sol^n Normalization is the process of organizing data to reduce redundancy (duplicate data) and improve data integrity. It makes structuring a database in such a way that dependencies are properly enforced by dividing large tables into smaller, related tables.

Q-10 What does a join in Mysql mean?

Soln In MySQL, a Join is used to combine rows from two or more tables based on a related column between them. It's a way to fetch data that is logically connected but stored in different tables.

Q-11 What do you understand about DDL, DCL and DML in MySQL?

Soln 1) DDL :- Data Definition Language.

- DDL commands are used to define and modify the structure of database objects like. tables, schema, indexes etc.
- Eg. Create, Alter, Drop

2) DML :- Data Manipulation Language:

- DML commands are used to manipulate the data stored in tables. The operations can be rollback if within a transaction
- Ex. Select, Insert, Update, Delete.

3] DCL :- Data Control language.

- DCL Commands are used to control access the data and database System.
- Ex. Grant, Revoke.

Q-12 What is the role of the MySQL JOIN clause in a query, and what are some common types of joins?

Soln Role of the join clause in a Query :-

- Combine related data from different tables.
- Allows you to follow relationship.
- Helps to maintain normalized database design without sacrificing query capability.
- Reduce data redundancy while still enabling complex data retrival.

→ Common Types of JOIN's in MySQL :-

1] INNER JOIN
2] LEFT Join
3] Right Join
4] FULL Outer Join
5] Cross Join