SVKM’s NMIMS
Mukesh Patel School of Technology Management & Engineering
A.Y. 2022 - 23
Course: Database Management Systems

Project Report

Program	BTech CSBS	
Semester	4	
Name of the Project:	Iron and Steel Trading Management System	
Details of Project Members		
Batch	**Roll No.**	**Name**
1	E004	Tanish Anam
Date of Submission: 04/04/2023		

Note:

1. Create a readme file if you have multiple files
2. All files must be properly named (I004_DBMSProject)
3. Submit all relevant files of your work (Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

Rubrics for the Project evaluation:

- Innovative Ideas and self learning (5 Marks) Idea should not be regular such as Hotel, Library Management system etc.
- Implementation and Design (10 Marks) It includes ER model, Relational model and Normalization of tables.
- Project Demonstration and Viva (5 Marks)

Project Report

Selected Topic

Iron and Steel Trading Management System

by

Tanish Anam, Roll number: E004

Course: DBMS

AY: 2022-23

Table of Contents

I. Storyline

(This section should describe the requirements for the chosen database topic. Form a storyline and describe in detail.)

An iron and steel trading management system is a critical tool for businesses that trade in these commodities.

To start with, the system should have an item management feature that can manage the inventory of various types of iron and steel products. This feature should allow users to store information such as product name, size, weight, and quantity. The item management feature should also provide real-time updates from suppliers and track the availability of products in the market.

The system should also have a trader management feature that enables users to maintain records of their suppliers and customers. The feature should allow users to store information such as contact details, order history, and special requirements of each trader.

Type your text

The factory management feature should allow users to store information on the factories where the iron and steel products are produced. This feature should also allow users to track the availability of products from each factory, as well as the quality of the products.

The bill management feature should enable users to create, manage, and track bills for products traded in the system. The feature should also allow users to manage payments and track outstanding bills.

The loader management feature should enable users to manage their fleet of vehicles that transport iron and steel products. This feature should allow users to store information such as vehicle type, registration number, and driver details.

The driver management feature should enable users to maintain records of their drivers, including their contact details and other relevant information.

The employee management feature should enable users to maintain records of their employees, including their contact details, name, ids and payroll information.

Finally, the system should have a dependent management feature that enables users to manage the dependents of their employees. This feature should allow users to store information such as the names, insurance number, and contact details of the dependents.

In summary, an iron and steel trading management system should have features such as item management, trader management, factory management, bill management, loader management, driver management, employee management, and dependent management. These features will enable users to effectively manage their business operations and make data-driven decisions to maximize their profitability.

II. Components of Database Design

1. **Items**: It is an entity which comprises all the items that a trader can sell or buy

   Attributes: Item ID, Weight, Availability, Type

2. **Trader:** This entity gives us information about the trader i.e. whether he is a buyer or seller and other vital information about the person

   Attributes: Trader ID, Location, Type, Name, Contact Number

3. **Factories:** This entity tells us about different factories present across India

   Attributes: Factory ID, Name, Stock, Contact Number, Location, Owner

4. **Drivers**: This entity lets us know about the details of the drivers who deliver the products from one location to another

   Attributes: Driver ID, Name, Salary, Contact Number

5. **Loaders:** They are the vehicles which are driven by drivers to deliver the orders

   Attributes: Loader ID, Type, Loader Number, End_Loc, Start_Loc, Total Days Taken, DriverID(FK)

6. **Bill:** It tells us about the total amount of the bill and its details

   Attributes: GSTIN, E-Invoice Number, Total Amount, Billing Name

7. **Employee:** This entity lets us know about the employees working currently

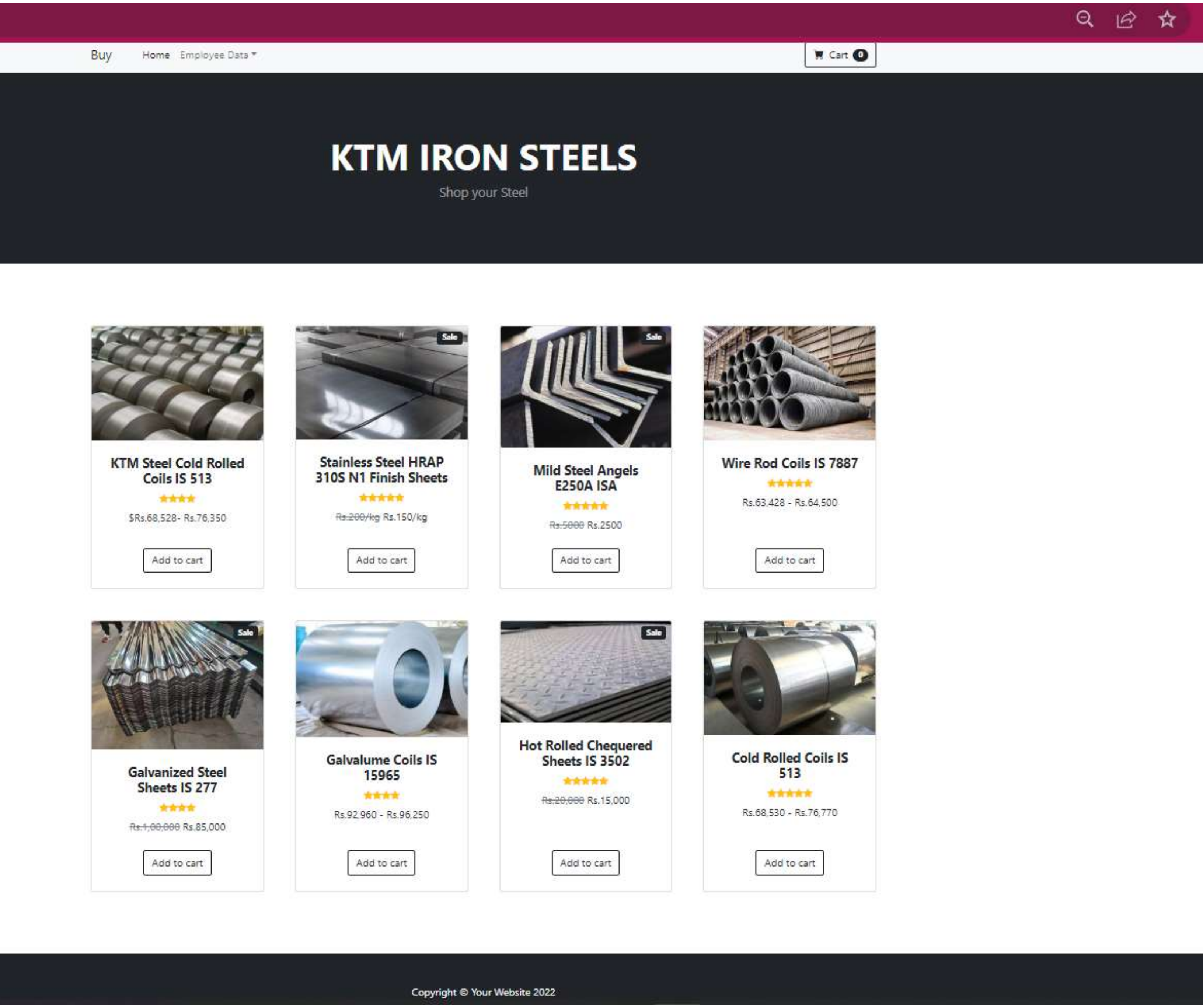
   Attributes: Employee ID, Company Name, Salary, Number, Name (First Name, Last_Name)

8. **Dependents:** This tells us about the dependents of the employees that work for our company

   Attributes: Name, Number, Insurance, Employee_ID (FK)

Relationships:

1. Sells/Buys: Traders and Items - One to Many
2. Orders: Traders and Drivers- One to Many
3. Drive: Drivers and Loaders- One to Many
4. Coordinates: Employee and Drivers: Many to Many
5. Works for: Trader to Employee: One to Many
6. Generates: Employee to Bill: One to One
7. Has: Employees to Dependents: One to Many
8. Buys from: Traders to Factories: One to Many



Buy Home Employee Data ▾ Cart 0

KTM IRON STEELS

Shop your Steel

Sale

Sale

Sale

Sale

KTM Steel Cold Rolled Coils IS 513
★★★★
$Rs.68,528- Rs.76,350
Add to cart

Stainless Steel HRAP 310S N1 Finish Sheets
★★★★★
~~Rs.200/kg~~ Rs.150/kg
Add to cart

Mild Steel Angels E250A ISA
★★★★★
~~Rs.5000~~ Rs.2500
Add to cart

Wire Rod Coils IS 7887
★★★★★
Rs.63,428 - Rs.64,500
Add to cart

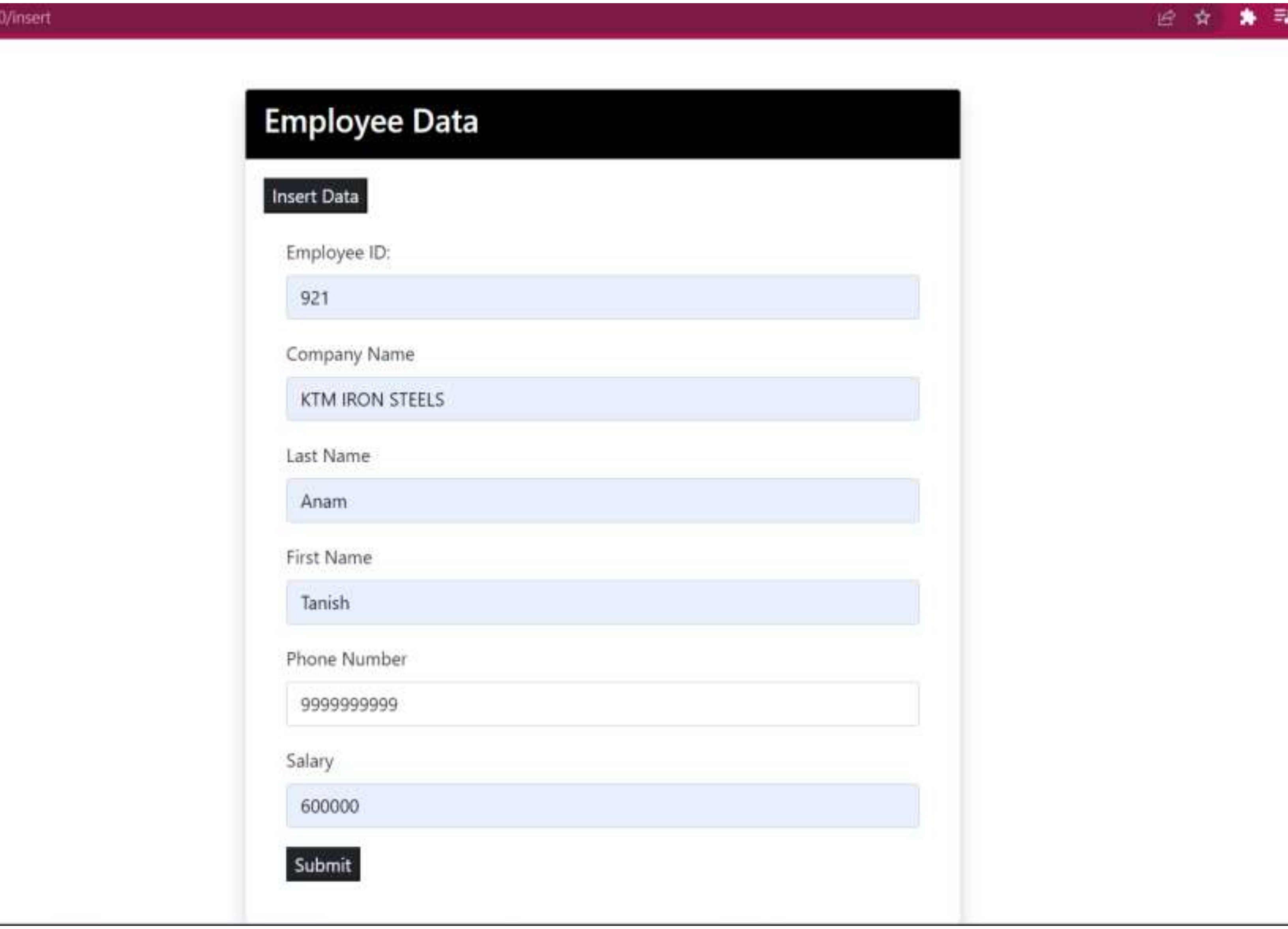
Galvanized Steel Sheets IS 277
★★★★
~~Rs.1,00,000~~ Rs.85,000
Add to cart

Galvalume Coils IS 15965
★★★★
Rs.92,960 - Rs.96,250
Add to cart

Hot Rolled Chequered Sheets IS 3502
★★★★★
~~Rs.20,000~~ Rs.15,000
Add to cart

Cold Rolled Coils IS 513
★★★★★
Rs.68,530 - Rs.76,770
Add to cart

Copyright © Your Website 2022



Employee Data

Employee ID: 921

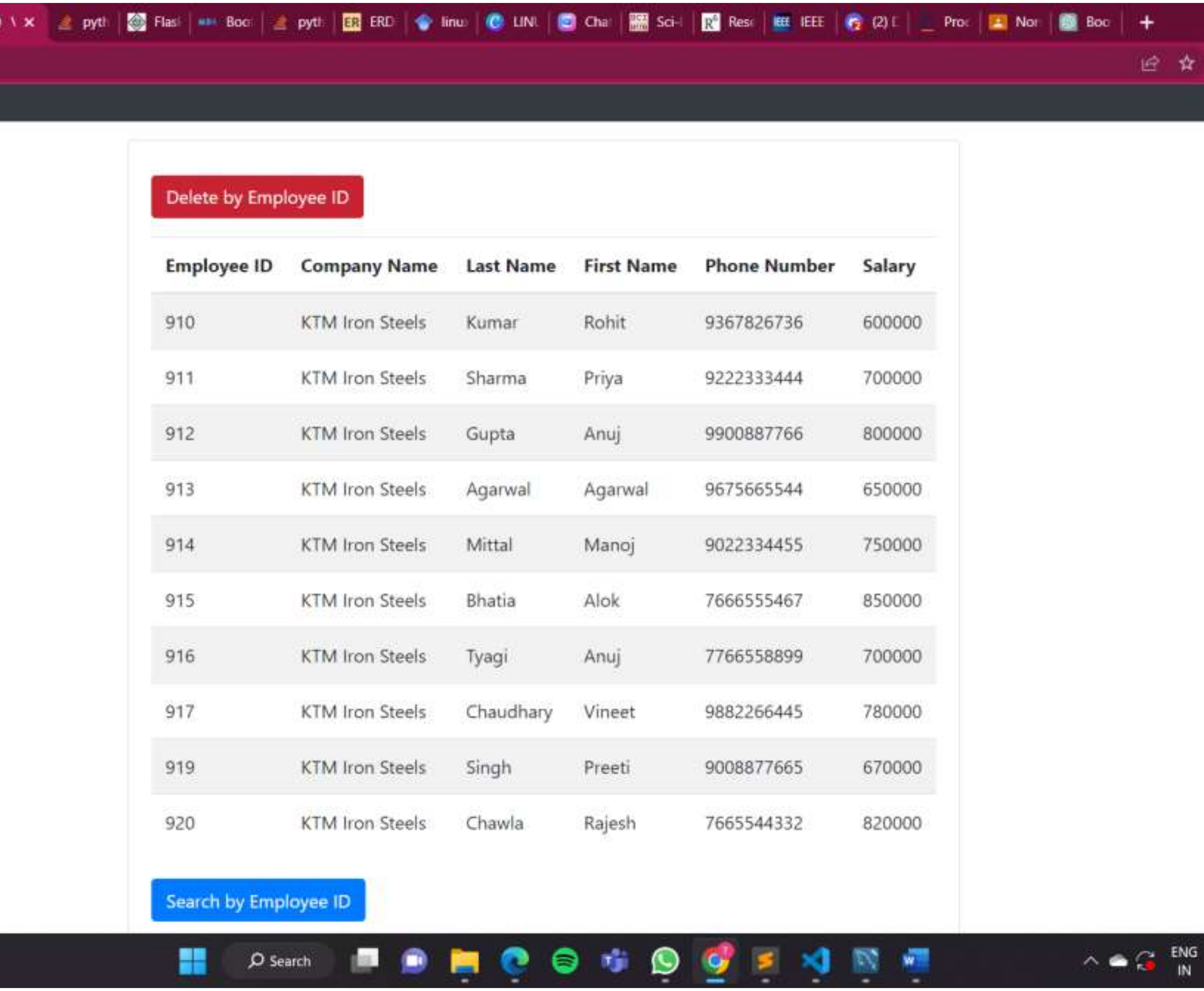
Company Name: KTM IRON STEELS

Last Name: Anam

First Name: Tanish

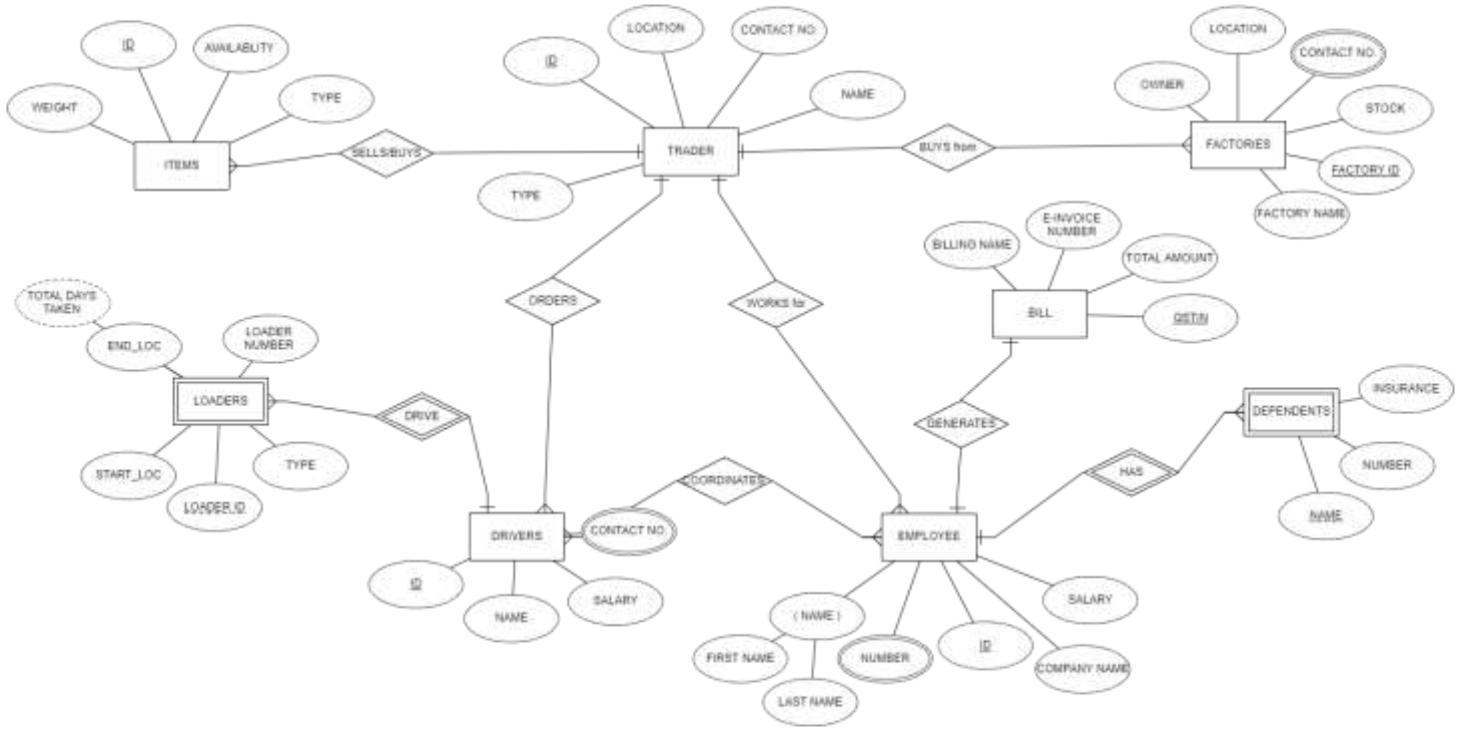
Phone Number: 9999999999

Salary: 600000



Employee ID	Company Name	Last Name	First Name	Phone Number	Salary
910	KTM Iron Steels	Kumar	Rohit	9367826736	600000
911	KTM Iron Steels	Sharma	Priya	9222333444	700000
912	KTM Iron Steels	Gupta	Anuj	9900887766	800000
913	KTM Iron Steels	Agarwal	Agarwal	9675665544	650000
914	KTM Iron Steels	Mittal	Manoj	9022334455	750000
915	KTM Iron Steels	Bhatia	Alok	7666555467	850000
916	KTM Iron Steels	Tyagi	Anuj	7766558899	700000
917	KTM Iron Steels	Chaudhary	Vineet	9882266445	780000
919	KTM Iron Steels	Singh	Preeti	9008877665	670000
920	KTM Iron Steels	Chawla	Rajesh	7665544332	820000

III. Entity Relationship Diagram



IV. Relational Model

Items (id, weight, availability, type)

Loaders (loader_id, end_loc, start_loc, type, loader_number, d_id*)

Drivers (d_id, name, salary, contact_no, e_id*)

Employee (e-id, e_name, number, first_name, last_name, salary, company_name)

Bill (gstin, billing_name, total_amount, e_invoice_number, e_id*)

Factories (factory_id, factory_name, stock, f_contact_no, location, owner)

Trader (t_id, location, t_contact_no, t_name, id*, d_id, e_id*, factory_id*)

Dependents (d_name, d_number, insurance, e_id*)

Drivers-contact_no (d_id, contact_no)

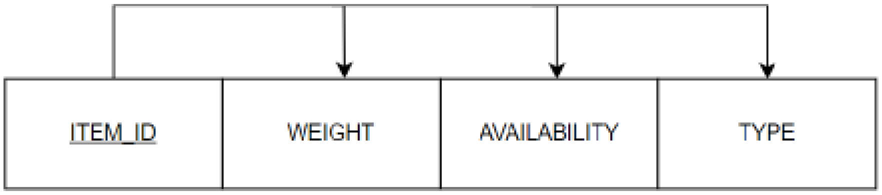
Employee-number (e_id, number)

Factories-f_contact_no (factory_id, f_contact_no)

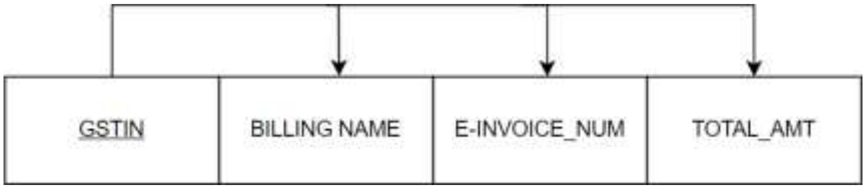
Trader-t_contact_no (t_id, t_contact_no)

V. Normalization

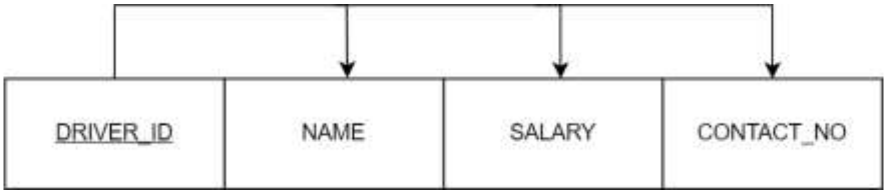
1. Item_ID --> Item_Availability, Item_Weight, Item_type



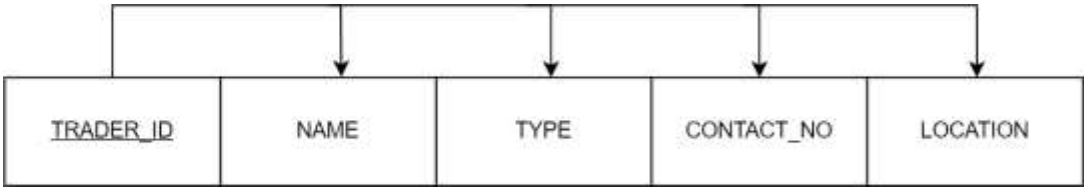
2. GSTIN --> Billing_name, E-Invoice, Total_amt



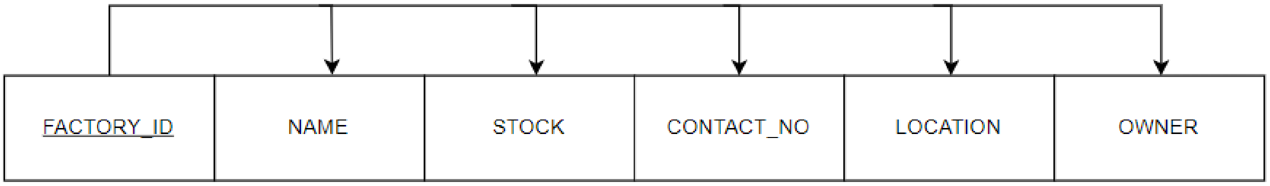
3. Driver_ID -->Driver_Name, Driver_Salary, Driver_Number



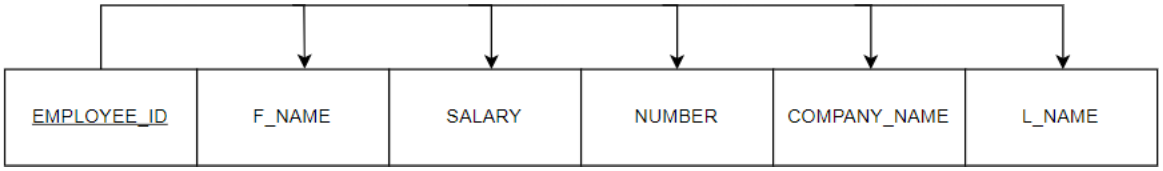
4. Trader_id--> trader_name, trader_type, trader_number, trader_location



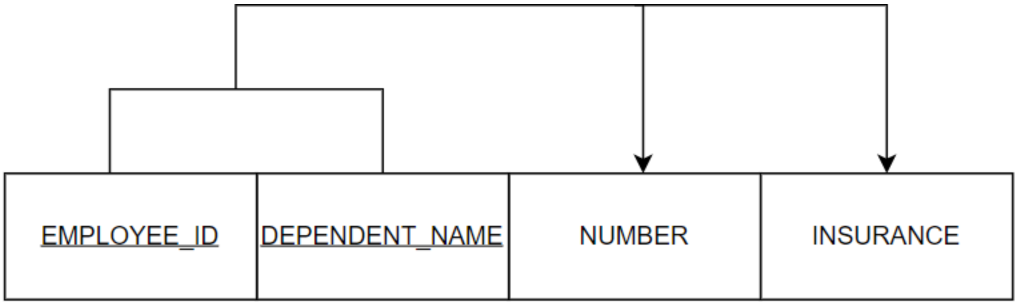
5. Factory_ID --> Fact_Name, Stock, Fact_Number, Fact_Location, Fact_Owner



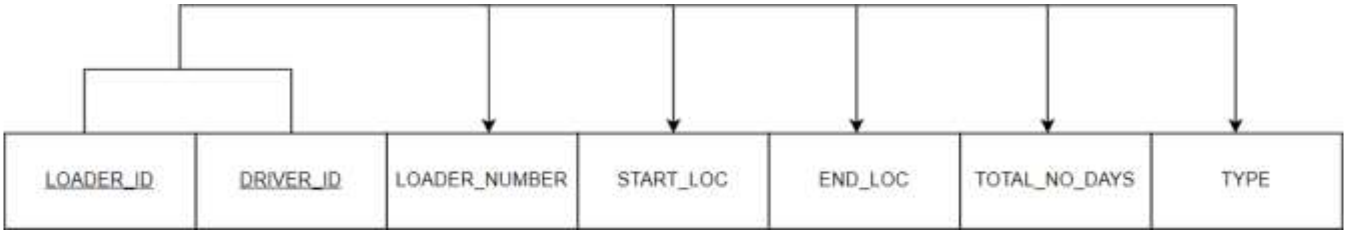
6. Employee_ID --> f_name, salary, number, company_name, l_name



7. Emp_ID, Depen_Name --> Depen_Number, Insurance



8. Loader_id, Driver_ID --> Loader_No, start_loc, end_loc, total_no_days, type



Our given relationships are in 1NF since:

a) There are no repeating groups in the table i.e., each row/ column intersection contains one and only one value, not a set of values (atomic in nature).
b) All the primary keys are identified for each relationship.
c) All attributes are dependent on the Primary key.

Our given relationships are also in 2NF since:

a) It is in 1NF.
b) There are no partial dependencies i.e., no attribute is dependent only on a portion of the primary key.

Our given relationships are also in 3NF since:

a) The relationship is already in 2NF as there are no partial dependencies present.
b) There exists No Transitive Dependencies i.e. No attribute functionally depends on another non-prime attribute.

Our given relationships are also in BCNF (Boyce-Codd NF) since:

a) It is in 3NF.
b) Every Determinant in the Table is a Candidate Key.

VI. SQL Queries

CREATING TABLES:

```
create database dbms_project;
show databases;
use dbms_project;

CREATE TABLE TRADERS (
TRADER_ID         VARCHAR(10) PRIMARY KEY,
TRADER_NAME VARCHAR(20) NOT NULL,
TRADER_CO_NUMBER NUMERIC(10),
TRADER_TYPE   VARCHAR(20) NOT NULL,
TRADER_LOCATION  VARCHAR(20) NOT NULL);

CREATE TABLE EMPLOYEEES (
EMP_ID                NUMERIC(10) PRIMARY KEY,
EMP_COMPANY_NAME        VARCHAR(20),
EMP_LNAME       VARCHAR(15)  NOT  NULL,
EMP_FNAME       VARCHAR(15)  NOT  NULL,
EMP_PHONE       VARCHAR(10)  NOT  NULL,
EMP_SALARY      NUMERIC(20));
ALTER TABLE EMPLOYEEES RENAME TO EMPLOYYEE;

CREATE TABLE FACTORYY (
FACTORY_ID       NUMERIC(10) PRIMARY KEY,
FACTORY_NAME VARCHAR(50) NOT NULL,
FACTORY_CO_NUMBER       NUMERIC(10),
FACTORY_STOCK        NUMERIC(50),
FACTORY_LOCATIONVARCHAR(20),
FACTORY_OWNER VARCHAR(20) );
ALTER TABLE FACTORYY RENAME TO FACTORY;

CREATE TABLE ITEMSSSSSS (
ITEM_ID     NUMERIC(10) PRIMARY KEY,
ITEM_AVAILABILITY VARCHAR(50),
ITEM_WEIGHT    NUMERIC(50),
ITEM_TYPE          VARCHAR(50) );
ALTER TABLE ITEMSSSSSS RENAME TO ITEM;
```

```
CREATE TABLE DRIVERS (
DRIVER_ID          NUMERIC(10) PRIMARY KEY,
DRIVER_CO_NUMBER          NUMERIC(10),
DRIVER_SALARY         NUMERIC(50) ,
DRIVER_NAME  VARCHAR(20) );

CREATE TABLE BILLS (
GSTIN         NUMERIC PRIMARY KEY,
E_INVOICE_NUMBER NUMERIC NOT NULL,
BILLING_NAME VARCHAR(20),
TOTAL_AMOUNT NUMERIC(50));

CREATE TABLE DEPENDENTSS(
EMP_ID                NUMERIC(10),
DEPENDENT_NAME VARCHAR(20) PRIMARY KEY,
DEPENDENT_NUMBER NUMERIC(10),
INSURANCE NUMERIC,
CONSTRAINT      DEPENDENT FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID));
ALTER TABLE DEPENDENTSS RENAME TO DEPENDENT;

CREATE TABLE LOADERSS (
DRIVER_ID          NUMERIC(10),
LOADER_ID NUMERIC(20) PRIMARY KEY,
LOADER_NUMBER VARCHAR(20),
START_LOC VARCHAR(20) NOT NULL,
END_LOC VARCHAR(20),
TOTAL_DAYS_TAKEN NUMERIC(20),
CONSTRAINT      LOADER FOREIGN KEY(DRIVER_ID) REFERENCES DRIVERS(DRIVER_ID));
ALTER TABLE LOADERSS RENAME TO LOADER;
```

Inserting Values:

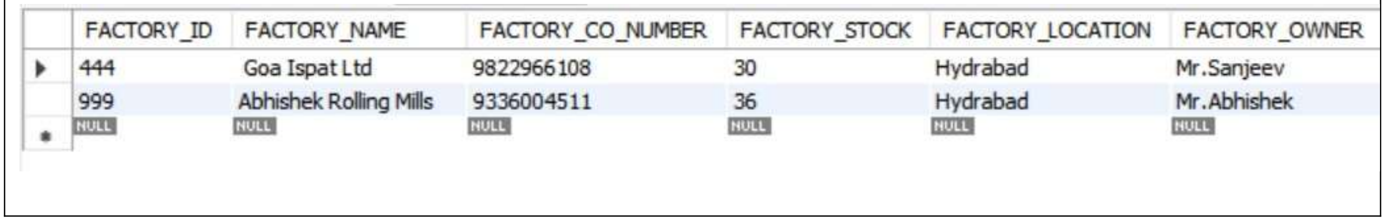
```
INSERT INTO TRADERS  VALUES
('123','Mr.Kanyum','9326656954','Buyer', 'Bhiwandi '),
('234','Mr.Sunil','9969443411','Seller', 'Mumbai '),
('345','Mrs.Lata','9008991009','Buyer', 'Salem'),
('456','Mr.Abhay ','9823620150','Seller', 'Goa'),
('567','Mr.Vishal','9923822122','Buyer', 'Kochin'),
('678','Mr.Babu ','9840739941','Seller', 'Trishur'),
('789','Mr.Biju','9447182897','Buyer', 'Goa'),
('891','Mr.Shreenivas','9032622719','Seller', 'Vishakapatnam'),
('910','Mr.Krishnaswamy ','9293713063','Buyer', 'Vijaywada'),
('911','Mr.Vinod','9843127400','Seller', 'Coimbatore');

select * FROM TRADERS;


INSERT INTO FACTORY VALUES
('111',' Shriram Steels','8461020301','22','Raipur', 'Mr.Rajesh'),
('222','Alankar Steels Pvt Ltd ','9827164726','25','Goa', 'Mr.Rajkumar'),
('333','Jagruti Industries Pvt LTd ','9494594008','27','Raipur', 'Mr.Sagar '),
('444',' Goa Ispat Ltd','9822966108','30','Hydrabad ', 'Mr.Sanjeev '),
('555','Ramsons Castings Pvt Ltd ','9822663313','34','Raipur', 'Mr.Rajesh '),
('666',' Balaji Industries Ltd','7803085135','37','Kolhapur', 'Mr.Tiwari '),
('777',' Saurabh Rolling Mills','9162667811','18','Goa', 'Mr.Mishra '),
('888','Arya Rolling Mills ','9225846008','40','Raipur', 'Mr.Alok '),
('999','Abhishek Rolling Mills ','9336004511','36','Hydrabad ', 'Mr.Abhishek '),
('101','Trimurti Industries ','9887851108','39','Raipur', 'Mr.Naresh ');

select * FROM FACTORY;

INSERT INTO EMPLOYYEE   VALUES
('910', 'KTM Iron Steels', 'Kumar ', 'Rohit ', '9367826736', '600000 '),
('911 ', 'KTM Iron Steels', 'Sharma ', 'Priya ', '9222333444', '700000 '),
('912 ', 'KTM Iron Steels', 'Gupta ', 'Anuj ', '9900887766', '800000 '),
('913 ', 'KTM Iron Steels', 'Agarwal ', 'Agarwal ', '9675665544', '650000 '),
('914 ', 'KTM Iron Steels', 'Mittal ', 'Manoj ', '9022334455', '750000 '),
('915 ', 'KTM Iron Steels', 'Bhatia ', 'Alok ', '7666555467', ' 850000 '),
('916', 'KTM Iron Steels', 'Tyagi ', 'Anuj ', '7766558899', '700000 '),
('917 ', 'KTM Iron Steels', 'Chaudhary ', 'Vineet ', '9882266445', ' 780000'),
('919 ', 'KTM Iron Steels', 'Singh ', 'Preeti ', '9008877665', ' 670000 '),
('920 ', 'KTM Iron Steels', 'Chawla ', 'Rajesh ', '7665544332', ' 820000 ');

select * FROM EMPLOYYEE;
```

```
INSERT INTO  ITEM  VALUES
('001 ', 'RPR  ', '25', 'KTM Steel Cold Rolled Coils IS 513'),
('002', 'Kolhapur ', '250', 'Stainless Steel HRAP 310S N1 Finish Sheets'),
('003', 'HYD ', '300', 'KTM Steel Cold Rolled Coils IS 513'),
('004', 'RPR  ', '550', 'Mild Steel Angels E250A ISA'),
('005 ', 'GOA ', '25', 'Wire Rod Coils IS 7887'),
('006 ', 'GOA ', '13', 'Wire Galvanized Steel Sheets IS 277'),
('007 ', 'HYD ', '75', 'Galvalume Coins IS 15965'),
('008 ', 'RPR ', '250', 'Hot Rolled Chequered Sheets IS 3502'),
('009 ', 'RPR ', '600', 'Cold Rolled Coils IS 513'),
('010 ', 'RPR  ', '75', 'Wire Rod Coils IS 7887');

select * FROM ITEM;


INSERT INTO DRIVERS  VALUES
('201 ', '9876543245 ', '25000 ', 'Vilas '),
('202 ', '9087346789 ', '27000 ', 'Rashid  '),
('203 ', '9123456789 ', '26000 ', 'Pankaj  '),
('204 ', '9458034612 ', '30000 ', 'Suraj  '),
('205 ', '9482947380 ', '34000 ', 'Suresh  '),
('206 ', '7654389076 ', '31000 ', 'Abdul '),
('207 ', '7056936213 ', '36000 ', 'Amir '),
('208 ', '7098765432 ', '28000 ', 'Paresh  '),
('209 ', '7260099887 ', '29000 ', 'Suresh  '),
('210 ', '9776644335 ', '33000 ', 'Ram ');

select * FROM DRIVERS;


INSERT INTO BILLS VALUES
('12', '11 ', 'Mr.Sanjeev ', '10000 '),
('23', '22 ', 'Mr.Alok ', '9000 '),
('34', '33 ', 'Mr.Naresh  ', '8500 '),
('45', '44 ', 'Mr.Rajesh ', '11000 '),
('56', '55 ', 'Mr.Shreenivas ', '9900 '),
('67', '66 ', 'Mr.Abhay  ', '8700 '),
('78', '77 ', 'Mr.Krishnaswamy  ', '11100 '),
('89', '88 ', 'Mr.Biju ', '9500 '),
('90', '99 ', 'Mr.Sunil ', '9300 '),
('24', '10 ', 'Mr.Vishal ', '7800 ');

select * FROM BILLS;


INSERT INTO DEPENDENT VALUES
('910 ', ' Naya', '9876543678', '111111111 '),
('911 ', 'Ishaan ', '9875900987', '222222222 '),
('912 ', ' Jai', '9345263546', '333333333 '),
('913 ', 'Inaya ', '7465365365', '444444444 '),
('914 ', 'Amar ', '7666555443', '555555555 '),
('915 ', 'Navi ', '9887766554', '666666666 '),
('916 ', 'Dhruv ', '7099876556', '777777777 '),
('917 ', ' Kanan', '9225533664', '888888888 '),
('919 ', 'Navya ', '9748950473', '999999999 '),
('920 ', 'Shrishti ', '7920283928', '101010101 ');

select * FROM DEPENDENT;


INSERT INTO LOADER  VALUES
('201 ', '301 ', 'MH 02 HK 2022', 'Mumbai ', 'Kolkata ', '4 '),
('202 ', '302 ', 'CG 11 QP 1023', 'Jaipur ', 'Chennai ', '5 '),
('203 ', '303 ', 'CG 18 HI 9012', 'Hyderabad ', 'Ahmedabad ', '6 '),
('204 ', '304 ', 'MH 02 KP 9900', 'Pune ', 'Chandigarh ', '5 '),
('205 ', '306 ', 'KA 21 WB 1231', 'Chennai ', 'Jaipur ', '4 '),
('206 ', '307 ', 'CG 18 BH 7641', 'Lucknow ', 'Kanpur ', ' 3'),
('207 ', '308 ', 'KA 02 LI 5655', 'Nagpur ', 'Indore ', ' 2'),
('208 ', '309 ', 'MH 03 KL 9232', 'Thane ', '        Bhopal ', '4 '),
('209 ', '305 ', 'KA 02 TH 8882', 'Visakhapatnam', 'Agra ', '5 '),
('210 ', '310 ', 'CG 17 GD 4521', 'Nashik ', 'Aurangabad ', '3 ');

select * FROM LOADER;
```

#1. Display information of Factory where the location is Hydrabad

```
select *
```

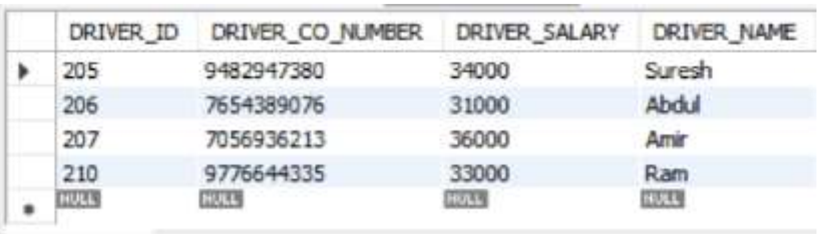
```
from factory
where factory_location="Hydrabad "
```



	FACTORY_ID	FACTORY_NAME	FACTORY_CO_NUMBER	FACTORY_STOCK	FACTORY_LOCATION	FACTORY_OWNER
▸	444	Goa Ispat Ltd	9822966108	30	Hydrabad	Mr.Sanjeev
	999	Abhishek Rolling Mills	9336004511	36	Hydrabad	Mr.Abhishek
*	NULL	NULL	NULL	NULL	NULL	NULL

#2. Display information of all drivers who's salary is greater than 30k

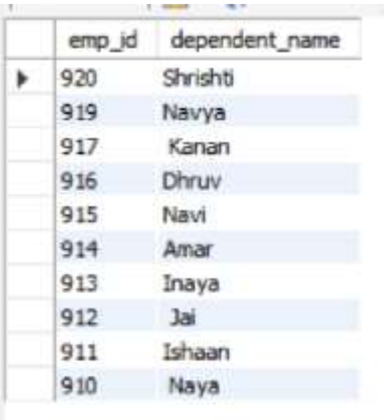
```
select *
from drivers
where driver_salary>30000
```



	DRIVER_ID	DRIVER_CO_NUMBER	DRIVER_SALARY	DRIVER_NAME
▸	205	9482947380	34000	Suresh
	206	7654389076	31000	Abdul
	207	7056936213	36000	Amir
	210	9776644335	33000	Ram
*	NULL	NULL	NULL	NULL

#3. Display the emp id and dependent name of all dependent in a descending order by the emp id

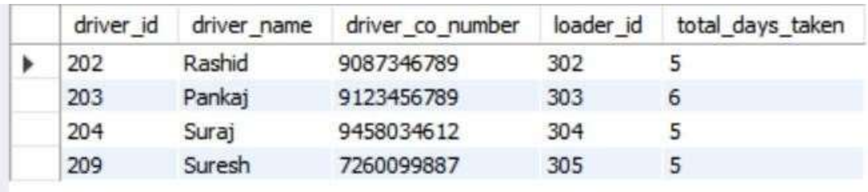
```
select emp_id, dependent_name
from dependent
order by emp_id desc
```



	emp_id	dependent_name
▸	920	Shrishti
	919	Navya
	917	Kanan
	916	Dhruv
	915	Navi
	914	Amar
	913	Inaya
	912	Jai
	911	Ishaan
	910	Naya

#4. Display the id, name, contact no., and total days take for those where the number of days is greater than or equal to 5

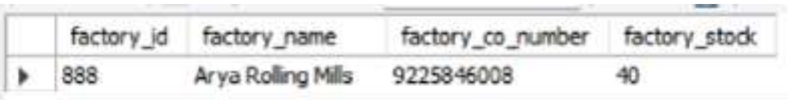
```
select drivers.driver_id, driver_name,driver_co_number,loader_id,total_days_taken
from drivers,loader
where drivers.driver_id=loader.driver_id and total_days_taken>=5
```



driver_id	driver_name	driver_co_number	loader_id	total_days_taken
202	Rashid	9087346789	302	5
203	Pankaj	9123456789	303	6
204	Suraj	9458034612	304	5
209	Suresh	7260099887	305	5

#5. Display id, name, contact no., and the number of stock for the factory which has the maximum no. of stock available

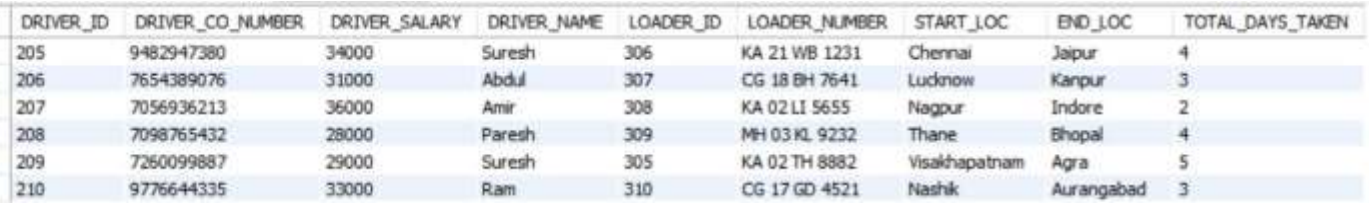
```
select factory_id, factory_name, factory_co_number, factory_stock
from factory
where factory_stock = (select max(factory_stock) from factory)
```



factory_id	factory_name	factory_co_number	factory_stock
888	Arya Rolling Mills	9225846008	40

#6. Display information of all drivers whose id comes in between 205 to 210 using inner join

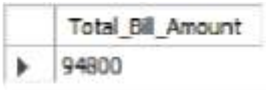
```
select *
from drivers natural inner join loader
where driver_id between ‘205’ and ‘210’
```



DRIVER_ID	DRIVER_CO_NUMBER	DRIVER_SALARY	DRIVER_NAME	LOADER_ID	LOADER_NUMBER	START_LOC	END_LOC	TOTAL_DAYS_TAKEN
205	9482947380	34000	Suresh	306	KA 21 WB 1231	Chennai	Jaipur	4
206	7654389076	31000	Abdul	307	CG 18 BH 7641	Lucknow	Kanpur	3
207	7056936213	36000	Amir	308	KA 02 LI 5655	Nagpur	Indore	2
208	7098765432	28000	Paresh	309	MH 03 KL 9232	Thane	Bhopal	4
209	7260099887	29000	Suresh	305	KA 02 TH 8882	Visakhapatnam	Agra	5
210	9776644335	33000	Ram	310	CG 17 GD 4521	Nashik	Aurangabad	3

#7. Write a query to find the total amount of bills generated

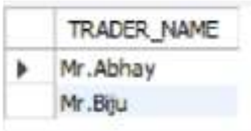
```
SELECT SUM(TOTAL_AMOUNT) as Total_Bill_Amount
FROM BILLS;
```



Total_Bill_Amount
94800

#8. Write a query to find the name of the traders whose location is 'Goa'

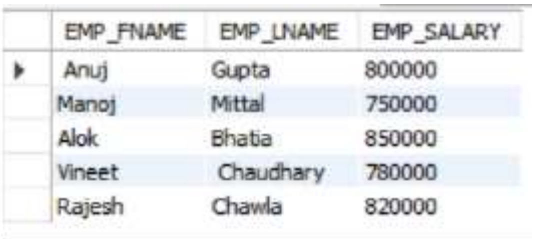
```
SELECT TRADER_NAME
FROM TRADERS
WHERE TRADER_LOCATION = 'Goa';
```



TRADER_NAME
Mr.Abhay
Mr.Biju

#9. Write a query to find the names and salaries of the employees whose salary is greater than 700000.

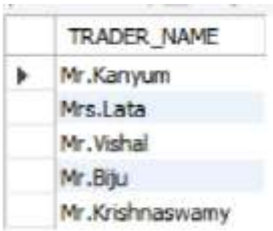
```
SELECT EMP_FNAME, EMP_LNAME, EMP_SALARY
FROM EMPLOYYEE
WHERE EMP_SALARY > 700000;
```



EMP_FNAME	EMP_LNAME	EMP_SALARY
Anuj	Gupta	800000
Manoj	Mittal	750000
Alok	Bhatia	850000
Vineet	Chaudhary	780000
Rajesh	Chawla	820000

#10.Write a query to find the traders who are also buyers.

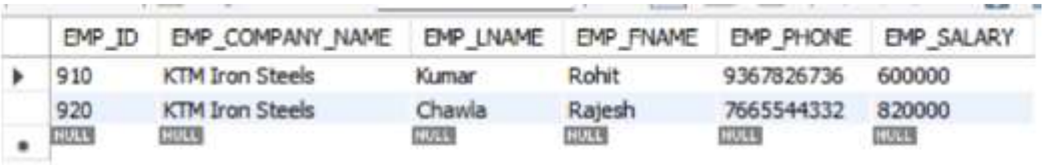
```
SELECT TRADER_NAME
FROM TRADERS
WHERE TRADER_TYPE = 'Buyer';
```



	TRADER_NAME
▶	Mr.Kanyum
	Mrs.Lata
	Mr.Vishal
	Mr.Biju
	Mr.Krishnaswamy

#11. Write a SQL query to select all employees whose first name starts with the letter 'R'.

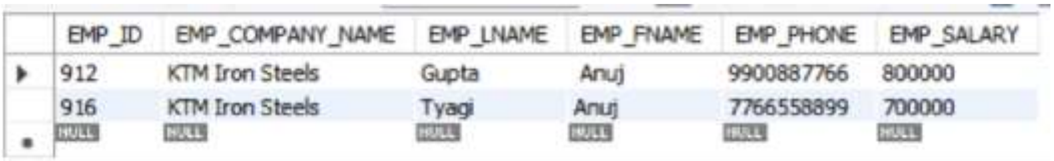
SELECT *
FROM EMPLOYYEE
WHERE EMP_FNAME LIKE 'R%';



	EMP_ID	EMP_COMPANY_NAME	EMP_LNAME	EMP_FNAME	EMP_PHONE	EMP_SALARY
▶	910	KTM Iron Steels	Kumar	Rohit	9367826736	600000
	920	KTM Iron Steels	Chawla	Rajesh	7665544332	820000
*	NULL	NULL	NULL	NULL	NULL	NULL

#12. Write a query where the first name of the employee is Anuj

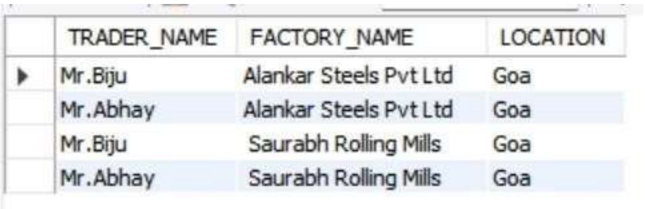
SELECT *
FROM EMPLOYYEE
WHERE EMP_FNAME like '%Anuj%';



	EMP_ID	EMP_COMPANY_NAME	EMP_LNAME	EMP_FNAME	EMP_PHONE	EMP_SALARY
▶	912	KTM Iron Steels	Gupta	Anuj	9900887766	800000
	916	KTM Iron Steels	Tyagi	Anuj	7766558899	700000
*	NULL	NULL	NULL	NULL	NULL	NULL

#13. Write an SQL query to retrieve the names of traders and factories located in the same city by taking the Cartesian product of TRADERS and FACTORY tables

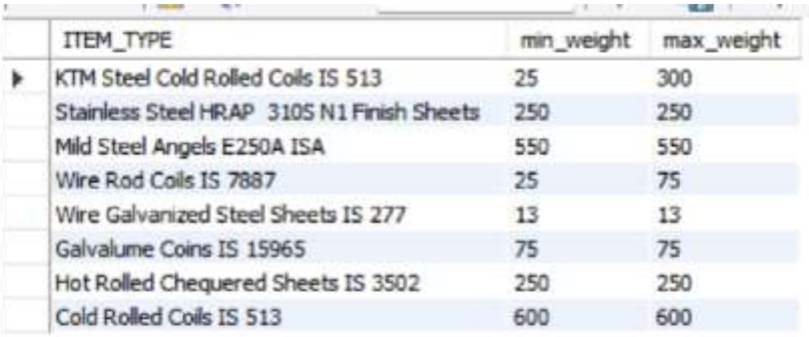
SELECT TRADER_NAME, FACTORY_NAME, TRADER_LOCATION as LOCATION
FROM TRADERS, FACTORY
WHERE TRADERS.TRADER_LOCATION = FACTORY.FACTORY_LOCATION;



	TRADER_NAME	FACTORY_NAME	LOCATION
▶	Mr.Biju	Alankar Steels Pvt Ltd	Goa
	Mr.Abhay	Alankar Steels Pvt Ltd	Goa
	Mr.Biju	Saurabh Rolling Mills	Goa
	Mr.Abhay	Saurabh Rolling Mills	Goa

#14. Write an SQL query to find the minimum and maximum weight of items for each type of item.

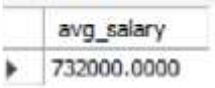
```
SELECT ITEM_TYPE, MIN(ITEM_WEIGHT) as min_weight, MAX(ITEM_WEIGHT)
as max_weight
FROM ITEM
GROUP BY ITEM_TYPE;
```



	ITEM_TYPE	min_weight	max_weight
▶	KTM Steel Cold Rolled Coils IS 513	25	300
	Stainless Steel HRAP 310S N1 Finish Sheets	250	250
	Mild Steel Angels E250A ISA	550	550
	Wire Rod Coils IS 7887	25	75
	Wire Galvanized Steel Sheets IS 277	13	13
	Galvalume Coins IS 15965	75	75
	Hot Rolled Chequered Sheets IS 3502	250	250
	Cold Rolled Coils IS 513	600	600

#15. Write an SQL query to find the average salary of all employees.

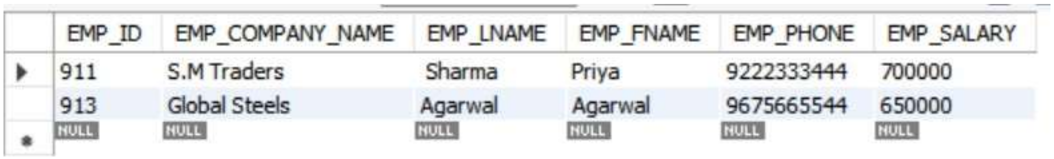
```
SELECT AVG(EMP_SALARY) as avg_salary
FROM EMPLOYEE;
```



	avg_salary
▶	732000.0000

#16. Write an SQL query to display the details of all the employees who have dependents and whose dependent names starts from an I

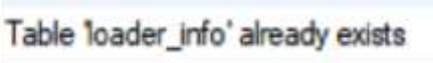
```
SELECT *
FROM EMPLOYEE
WHERE EMP_ID IN (
    SELECT EMP_ID
    FROM DEPENDENT
    WHERE DEPENDENT_NAME like "I%"
);
```



EMP_ID	EMP_COMPANY_NAME	EMP_LNAME	EMP_FNAME	EMP_PHONE	EMP_SALARY
911	S.M Traders	Sharma	Priya	9222333444	700000
913	Global Steels	Agarwal	Agarwal	9675665544	650000
NULL	NULL	NULL	NULL	NULL	NULL

#17. Create a view that displays the list of loaders with their respective driver names, the start location, and the total number of days taken.

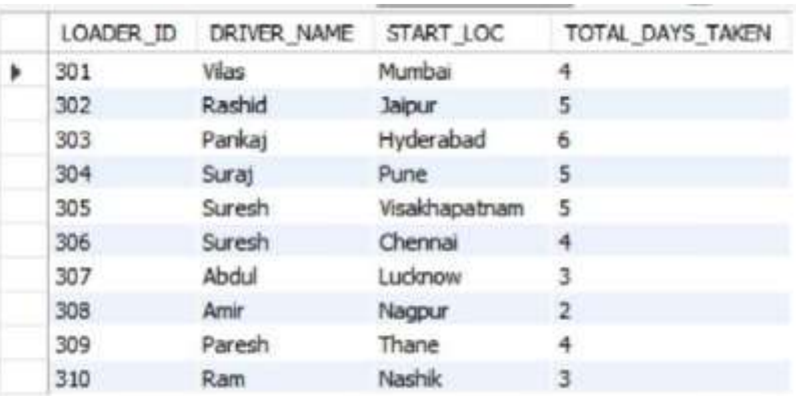
```
CREATE VIEW loader_info AS
SELECT LOADER.LOADER_ID, DRIVERS.DRIVER_NAME, LOADER.START_LOC,
LOADER.TOTAL_DAYS_TAKEN
FROM LOADER Inner JOIN DRIVERS
ON LOADER.DRIVER_ID = DRIVERS.DRIVER_ID;
```



```
Table 'loader_info' already exists
```

#18. Display the view loaders_info created above

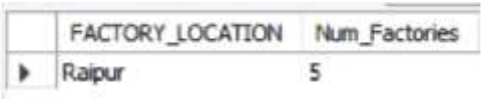
```
select *
from loader_info
```



LOADER_ID	DRIVER_NAME	START_LOC	TOTAL_DAYS_TAKEN
301	Vilas	Mumbai	4
302	Rashid	Jaipur	5
303	Pankaj	Hyderabad	6
304	Suraj	Pune	5
305	Suresh	Visakhapatnam	5
306	Suresh	Chennai	4
307	Abdul	Lucknow	3
308	Amir	Nagpur	2
309	Paresh	Thane	4
310	Ram	Nashik	3

#19. Find all cities with more than 2 factories in the same city

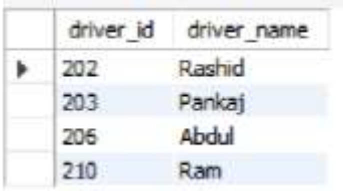
```
select FACTORY_LOCATION, count(*) as Num_Factories
from Factory
group by FACTORY_LOCATION
having count(*) > 2;
```



	FACTORY_LOCATION	Num_Factories
▶	Raipur	5

#20. Display the driver id and name who driver loaders which are Chhattisgarh using nested subqueries

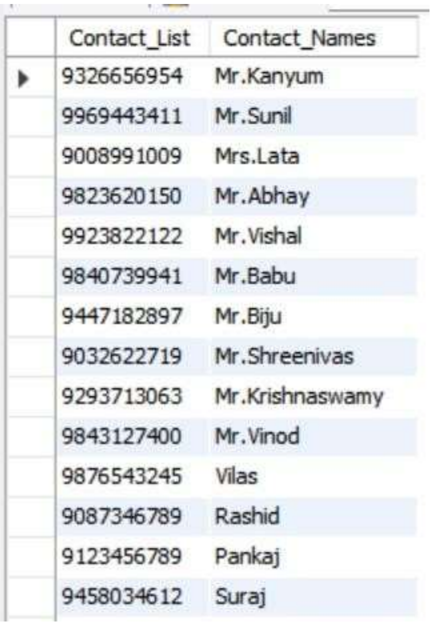
```
Select driver_id , driver_name
from drivers
where driver_id in (
select driver_id
from loader
where LOADER_NUMBER like "CG%"
);
```

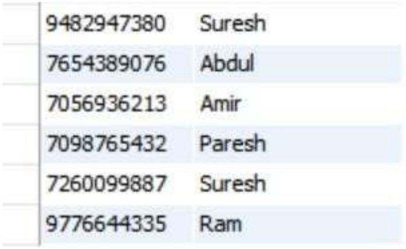


	driver_id	driver_name
▶	202	Rashid
	203	Pankaj
	206	Abdul
	210	Ram

#21. Display contact list of all drivers and traders using set operations

```
(select TRADER_CO_NUMBER as Contact_List ,TRADER_NAME as Contact_Names
from traders)
union
(select DRIVER_CO_NUMBER,DRIVER_NAME from drivers)
```

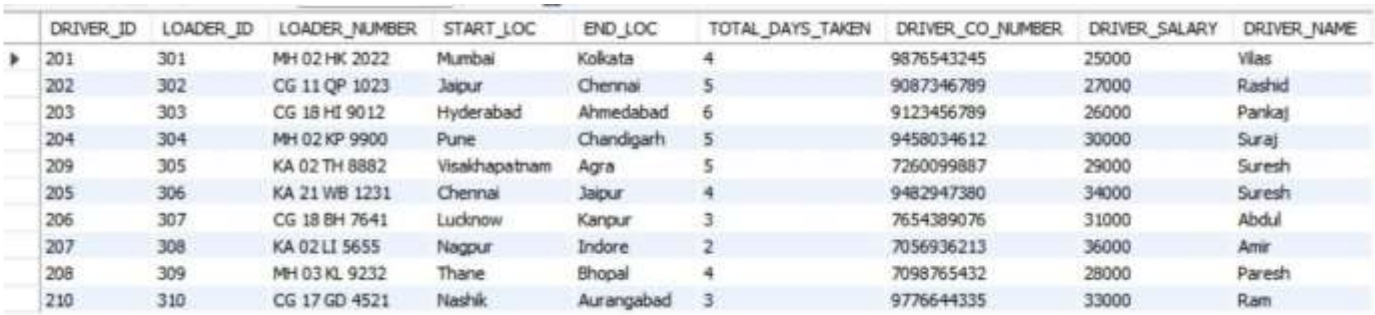




Contact_List	Contact_Names
9326656954	Mr.Kanyum
9969443411	Mr.Sunil
9008991009	Mrs.Lata
9823620150	Mr.Abhay
9923822122	Mr.Vishal
9840739941	Mr.Babu
9447182897	Mr.Biju
9032622719	Mr.Shreenivas
9293713063	Mr.Krishnaswamy
9843127400	Mr.Vinod
9876543245	Vilas
9087346789	Rashid
9123456789	Pankaj
9458034612	Suraj
9482947380	Suresh
7654389076	Abdul
7056936213	Amir
7098765432	Paresh
7260099887	Suresh
9776644335	Ram

#22.Write a query to Perform inner join on loader and drivers

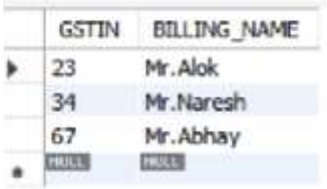
```
select *
from loader inner join drivers
using (driver_id);
```



DRIVER_ID	LOADER_ID	LOADER_NUMBER	START_LOC	END_LOC	TOTAL_DAYS_TAKEN	DRIVER_CO_NUMBER	DRIVER_SALARY	DRIVER_NAME
201	301	MH 02 HK 2022	Mumbai	Kolkata	4	9876543245	25000	Vilas
202	302	CG 11 QP 1023	Jaipur	Chennai	5	9087346789	27000	Rashid
203	303	CG 18 HI 9012	Hyderabad	Ahmedabad	6	9123456789	26000	Pankaj
204	304	MH 02 KP 9900	Pune	Chandigarh	5	9458034612	30000	Suraj
209	305	KA 02 TH 8882	Visakhapatnam	Agra	5	7260099887	29000	Suresh
205	306	KA 21 WB 1231	Chennai	Jaipur	4	9482947380	34000	Suresh
206	307	CG 18 BH 7641	Lucknow	Kanpur	3	7654389076	31000	Abdul
207	308	KA 02 LI 5655	Nagpur	Indore	2	7056936213	36000	Amir
208	309	MH 03 KL 9232	Thane	Bhopal	4	7098765432	28000	Paresh
210	310	CG 17 GD 4521	Nashik	Aurangabad	3	9776644335	33000	Ram

#23. Write a query to display Name and GSTIN whos billing amount is between 8k to 9k

```
select GSTIN,BILLING_NAME
from Bills
where TOTAL_AMOUNT between 8000 and 9000;
```



GSTIN	BILLING_NAME
23	Mr.Alok
34	Mr.Naresh
67	Mr.Abhay
NULL	NULL

VII. Project demonstration

- Tools/software/ libraries used
- Screenshot and Description of the Demonstration of project (if GUI is made)

- ERD Plus – To make the ERD Diagram.
- Draw.io - To make Dependency Diagram.
- MySQL – To make tables and store data in them.
- VSC – To write python code.
- Flask – For database connectivity using python.
- Sublime Text – For writing HTML codes.
- Bootstrap – For HTML templates.

Libraries Used

- Request
- Render template
- Flask

VIII. Learning from the Project

- How did this project help you?

This project has helped us increase our knowledge about MySQL and databases in general. It also helped us know how vast their applications are. We got the opportunity to learn how to create databases and tables and how to populate them with data to run queries currently and even in future. Working on a project like this helped us develop skills in database design and management, software development, data analysis, and project management.

- What new aspects did you learn?

In this project we have learnt Web Application Development using Flask and Database connectivity using Python.

IX. Challenges Faced

- Flask Connectivity
- Altering of Table values
- Technical difficulties with MySQL

X. Conclusion

- What are the key takeaways from the project?

Effective database management is critical for the success of any business, including those in the iron and steel trading industry. A well-designed SQL database management system can help organize and streamline operations, improve data accuracy, and enable more efficient decision-making. Also, Proper planning and requirements gathering are essential for developing a successful database management system.