



PROJECT FOR SQL MODULE

MILK DAIRY MANAGEMENT SYSTEM

Institute Name:- Itvedant Education Pvt .Ltd

Name:- HARSHAL SANDIP SHIRSEKAR

Email Address:- harryshirsekar@gmail.com

Date of submission:-26 May 2024

Project Aims:

Database Design and Schema Creation:

Create an optimized and normalized database schema that accurately represents the relationships between different entities such as Dairy Products, Factory Workers, distributors, and Customers.

Data Population and Integrity

Populate the database with sample data to simulate real-world scenarios in a Milk Dairy Management.

Ensure data integrity through the use of constraints, such as primary keys, foreign keys, and unique constraints.

User Authentication and Authorization:

Implement a user authentication system to secure the database, allowing only authorized users to access and modify data.

Define and enforce role-based access control to restrict users' actions based on their roles (e.g., Factory Team Leader, Administrator).

Transaction Management:

Implement a transaction management system to handle the distribution and product requirement tracking, etc...

Track transaction details, including dates, amount of distribution customers, and requirements of distribution customers require in the future.

Customer Requirement Tracking

Develop a mechanism for customers requiring issues on real real-time basis. (e.g., If customers require them to increase the amount of distribution they currently have daily, they can upgrade the amount of distribution as they need.)

Customer Information Management:

Create a system for managing customer information, including details such as names, contact information, and requirements of product types.

Implement features for adding, updating, and deactivating member records.

Reporting and Analysis:

Develop SQL queries for generating various reports, such as a list of currently distributed product details, pending amounts of customers, most products buy customers, etc

Implement analysis queries to gather insights into library usage patterns.

User-Friendly Interface:

Consider the development of a user-friendly interface or integrate the database with an existing milk dairy management system for seamless interaction.

Scalability and Performance:

Design the database with scalability in mind to accommodate potential future expansions of the Milk Dairy collection and user base.

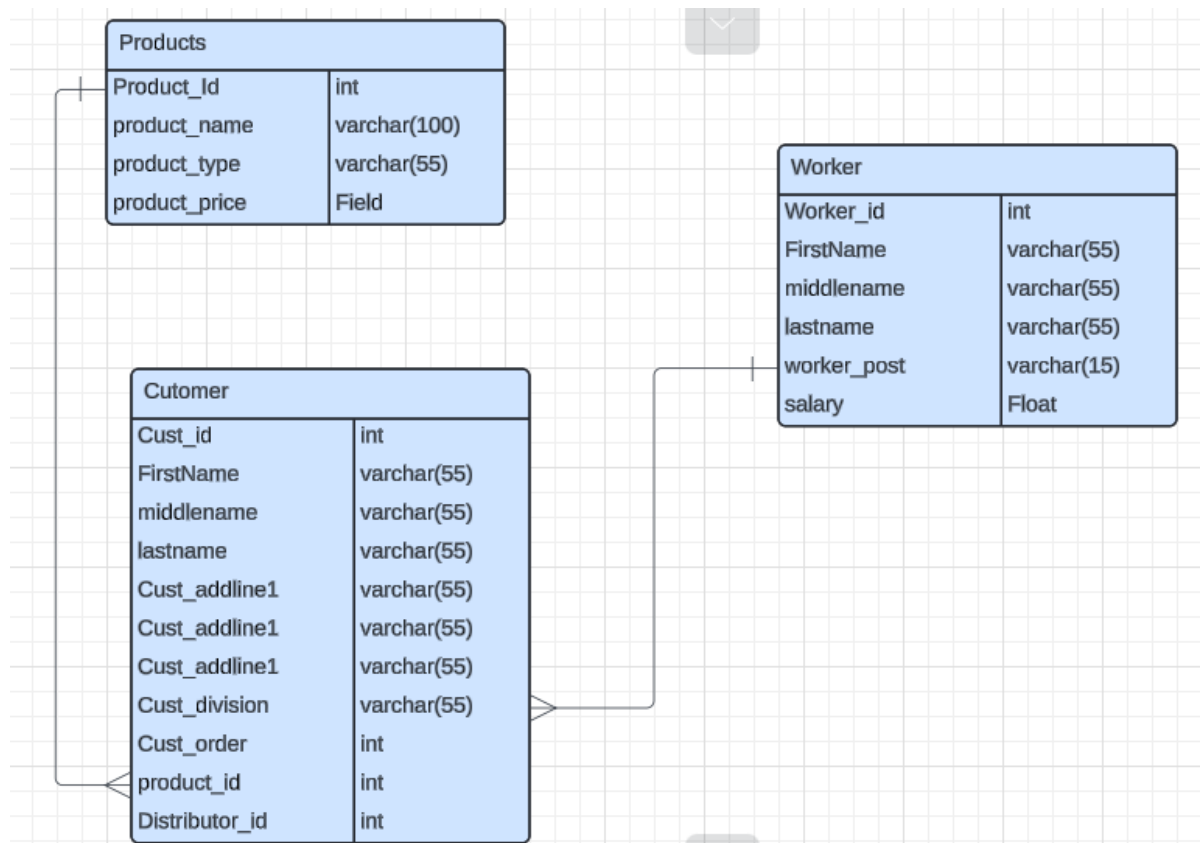
Optimize SQL queries and database indexing for improved performance.

By achieving these aims, the project aims to create a well-organized, secure, and efficient database system that enhances the overall management of the library's resources and services.

Project Objective:

The objective of this SQL project is to design and implement a comprehensive database system for a milk dairy management system. The primary focus is on creating a robust and efficient database schema to manage various aspects of dairy management operations, including product storage, customer information, worker information, customer requirements, and tracking monthly distribution. The project aims to enhance the overall efficiency and organization of the dairy by providing a centralized and well-structured database.

ER diagram:



Each rectangle represents an entity. Lines between the entities represent the relationships between them. The lines between the entities represent primary keys and foreign keys connecting the tables. The ER diagram showcases the relationships between the different tables, such as vehicle_details, accident_details, road_details, insurance_details, contractor_details as described in the project description.

Table description:

1: Product Table

Field	Type	Null	Key	Default	Extra
Product_Id	int	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
product_type	varchar(55)	YES		NULL	
product_price	decimal(4,2)	YES		NULL	

2: Worker Table

Field	Type	Null	Key	Default	Extra
Worker_id	int	NO	PRI	NULL	auto_increment
FirstName	varchar(55)	YES		NULL	
middlename	varchar(55)	YES		NULL	
lastname	varchar(55)	YES		NULL	
worker_post	varchar(15)	YES		NULL	
salary	float	NO		NULL	

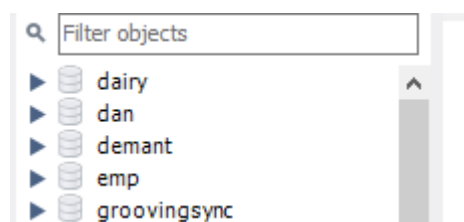
3: Customer Table

Field	Type	Null	Key	Default	Extra
Cust_id	int	NO	PRI	NULL	auto_increment
cust_fname	varchar(55)	YES		NULL	
cust_mname	varchar(55)	YES		NULL	
cust_lname	varchar(55)	YES		NULL	
Cust_addline1	varchar(55)	YES		NULL	
Cust_addline2	varchar(55)	YES		NULL	
Cust_addline3	varchar(55)	YES		NULL	
Cust_division	varchar(55)	YES		NULL	
Cust_order	int	NO		NULL	
product_id	int	YES	MUL	NULL	
Distributor_id	int	YES	MUL	NULL	

Commands:

1: Create Database-

CREATE DATABASE DAIRY;



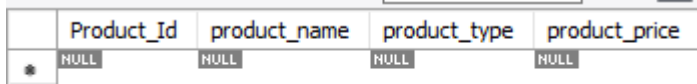
2: Using or Entering Database -

USE DAIRY;



3: Creating Tables -

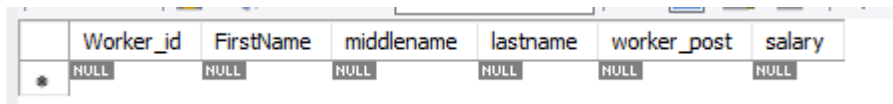
```
CREATE TABLE PRODUCTS(  
PRODUCT_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
PRODUCT_NAME VARCHAR(100),  
PRODUCT_TYPE VARCHAR (55),  
PRODUCT_PRICE DECIMAL (4,2));
```



A screenshot of a database table structure for the 'PRODUCTS' table. The table has four columns: 'Product_Id', 'product_name', 'product_type', and 'product_price'. Each column is shown with a 'NULL' value in a separate row below the header. The 'Product_Id' column is marked with a primary key icon (a small circle with a dot).

Product_Id	product_name	product_type	product_price
NULL	NULL	NULL	NULL

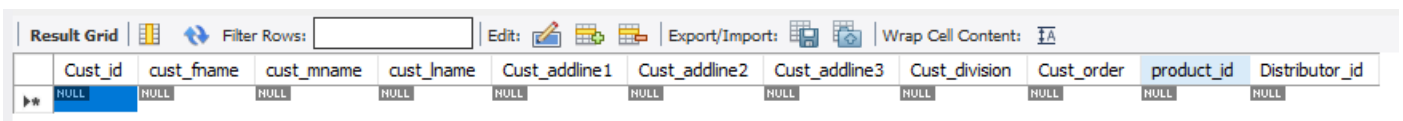
```
CREATE TABLE WORKER(  
WORKER_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
FIRSTNAME VARCHAR (55),  
MIDDLENAME VARCHAR (55),  
LASTNAME VARCHAR (55),  
WORKER_POST VARCHAR(15),  
SALARY FLOAT NOT NULL);
```



A screenshot of a database table structure for the 'WORKER' table. The table has six columns: 'Worker_id', 'FirstName', 'middlename', 'lastname', 'worker_post', and 'salary'. Each column is shown with a 'NULL' value in a separate row below the header. The 'Worker_id' column is marked with a primary key icon (a small circle with a dot).

Worker_id	FirstName	middlename	lastname	worker_post	salary
NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE CUSTOMER(  
CUST_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
CUST_FNAME VARCHAR (55),  
CUST_MNAME VARCHAR (55),  
CUST_LNAME VARCHAR (55),  
CUST_ADDLINE1 VARCHAR (55),  
CUST_ADDLINE2 VARCHAR (55),  
CUST_ADDLINE3 VARCHAR (55),  
CUST_DIVISION VARCHAR (55),  
CUST_ORDER INT NOT NULL,  
PRODUCT_ID INT,  
DISTRIBUTOR_ID INT,  
FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCTS (PRODUCT_ID),  
FOREIGN KEY (DISTRIBUTOR_ID) REFERENCES WORKER (WORKER_ID));
```



A screenshot of a database table structure for the 'CUSTOMER' table. The table has eleven columns: 'Cust_id', 'cust_fname', 'cust_mname', 'cust_lname', 'Cust_addline1', 'Cust_addline2', 'Cust_addline3', 'Cust_division', 'Cust_order', 'product_id', and 'Distributor_id'. Each column is shown with a 'NULL' value in a separate row below the header. The 'Cust_id' column is marked with a primary key icon (a small circle with a dot). The 'product_id' and 'Distributor_id' columns are highlighted in blue, indicating foreign key relationships.

Cust_id	cust_fname	cust_mname	cust_lname	Cust_addline1	Cust_addline2	Cust_addline3	Cust_division	Cust_order	product_id	Distributor_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4: Show Tables & Database-

```
SHOW DATABASES;
```

Result Grid	Filter Rows:
Database	
▶ dairy	
dan	
demant	
emp	
groovingsync	
index_demo	
information_schema	





SHOW TABLES;

Result Grid	Filter Rows:
Tables_in_dairy	
▶ customer	
products	
worker	

5: INSERTING DATA –

Product Table Entry:

```
INSERT INTO Products (Product_Id, product_name, product_type, product_price) VALUES
(001, "Cow Milk","1 Liter", 25),
(002, "Buffalo Milk","1 Liter", 35);
```

Result Grid		 Filter Rows:	<input type="text"/>	Edit:		
	Product_Id	product_name	product_type	product_price		
▶	1	Cow Milk	1 Liter	25.00		
	2	Buffalo Milk	1 Liter	35.00		
⌵	NULL	NULL	NULL	NULL		

Worker Table Entry:

```
INSERT INTO WORKER (WORKER_ID, FIRSTNAME, MIDDLENAME, LASTNAME, WORKER_POST,
SALARY) VALUES
(1001,"JAYRAM","PRAKASH","YADAV","MANAGER", 25000),
(1002,"RAJA","RAOJI","YADAV","HELPER", 20000),
(1003,"PRAKASH","SADASHIV","MORE","DISTRIBUTOR", 20000),
(1004,"KIRAN","SADASHIV","MORE","DISTRIBUTOR", 20000);
```

Result Grid

Filter Rows:

Edit:

Export

	Worker_id	FirstName	middlename	lastname	worker_post	salary
▶	1001	JAYRAM	PRAKASH	YADAV	Manager	25000
	1002	RAJA	RAOJI	YADAV	Helper	20000
	1003	PRAKASH	SADASHIV	MORE	Distributor	20000
	1004	KIRAN	SADASHIV	MORE	Distributor	20000
✱	NULL	NULL	NULL	NULL	NULL	NULL

Customer Table Entry:

```
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_MNAME, CUST_LNAME, CUST_ADDLINE1, CUST_ADDLINE2, CUST_ADDLINE3, CUST_DIVISION, CUST_ORDER, PRODUCT_ID, DISTRIBUTOR_ID) VALUES
```

```
(1,"GANGU"," ","RANE","ROOM NO 01","GANESH APT B WING","DEVENDRA INDUSTRY","SECTOR 1","2", 001,1003),
(2,"RAGHU","G","MORE","ROOM NO 02","GANESH APT B WING","DEVENDRA INDUSTRY","SECTOR 1","1", 002,1003),
(3,"JAYA","H","GHARE","ROOM NO 12","GANESH APT A WING","DEVENDRA INDUSTRY","SECTOR 1","3", 001, 1003),
(4,"MANU","M","GHOLE","ROOM NO 03","GANESH APT A WING","DEVENDRA INDUSTRY","SECTOR 1","3", 001, 1003),
(5,"RAJU","J","R","ROOM NO 01","SAI DARSHAN APT A WING","SURESH MEDICAL","SECTOR 3","2", 002, 1004),
(6,"JYOTI","S","JAGDALE","ROOM NO 02","SAI DARSHAN APT B WING","SURESH MEDICAL","SECTOR 3","2", 001,1004),
(7,"KIRAN","J","MHATRE","ROOM NO 14","SAI DARSHAN APT B WING","SURESH MEDICAL","SECTOR 3","2", 001,1004),
(8,"BAJI","SHAMBHU","BHOSALE","ROOM NO 22","SAI DARSHAN APT B WING","SURESH MEDICAL","SECTOR 3","3", 001,1004);
```

Result Grid											
Filter Rows: <input type="text"/>											
Edit: Export/Import: Wrap Cell Content:											
	Cust_id	cust_fname	cust_mname	cust_lname	Cust_addline1	Cust_addline2	Cust_addline3	Cust_division	Cust_order	product_id	Distributor_id
▶	1	Gangu		Rane	room no 01	Ganesh Apt B wing	Devendra Industry	Sector 1	2	1	1003
	2	Raghu	G	More	room no 02	Ganesh Apt B wing	Devendra Industry	Sector 1	1	2	1003
	3	Jaya	H	Ghare	room no 12	Ganesh Apt A wing	Devendra Industry	Sector 1	3	1	1003
	4	Manu	M	Ghole	room no 03	Ganesh Apt A wing	Devendra Industry	Sector 1	3	1	1003
	5	Raju	J	R	room no 01	Sai Darshan Apt A wing	Suresh Medical	Sector 3	2	2	1004
	6	Jyoti	S	Jagdale	room no 02	Sai Darshan Apt B wing	Suresh Medical	Sector 3	2	1	1004
	7	Kiran	J	Mhatre	room no 14	Sai Darshan Apt B wing	Suresh Medical	Sector 3	2	1	1004
	8	Baji	Shambhu	Bhosale	room no 22	Sai Darshan Apt B wing	Suresh Medical	Sector 3	3	1	1004
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6: DELETE TABLE –

```
DROP TABLE PRODUCT;
DROP TABLE WORKER;
DROP TABLE CUSTOMER;
```

7: TRUNCATE TABLE –

```
TRUNCATE PRODUCT;
TRUNCATE CUSTOMER;
TRUNCATE WORKER;
```

8: ALTER QUERY –

Add city in worker table.

```
ALTER TABLE WORKER ADD COLUMN CITY;
```


9: UPDATE QUERY –

Update salary 22000 for worker id 1002.

```
UPDATE WORKER SET SALARY = 22000 WHERE WORKER_ID = 1002;
```

	Worker_id	FirstName	middleName	lastName	POST	salary
▶	1001	JAYRAM	PRAKASH	YADAV	Manager	25000
	1002	RAJA	RAOJI	YADAV	Helper	22000
	1003	PRAKASH	SADASHIV	MORE	Distributor	20000
	1004	KIRAN	SADASHIV	MORE	Distributor	20000

10: DELETE QUERY –

Delete customer details where order is 1.

```
DELETE FROM CUSTOMER WHERE CUST_ORDER = 1;
```

Delete all customer details where order is greater than 2.

```
DELETE FROM CUSTOMER WHERE CUST_ORDER > 2;
```

10: TABLE DESCRIPTION –

DESC WORKER;

Field	Type	Null	Key	Default	Extra
Worker_id	int	NO	PRI	NULL	auto_increment
FirstName	varchar(55)	YES		NULL	
middleName	varchar(55)	YES		NULL	
lastName	varchar(55)	YES		NULL	
POST	varchar(55)	YES		NULL	
salary	float	NO		NULL	

11: SELECT QUERY –

```
SELECT * FROM PRODUCTS;
```

Product_Id	product_name	product_type	product_price
1	Cow Milk	1 Liter	25.00
2	Buffalo Milk	1 Liter	35.00
NULL	NULL	NULL	NULL

```
SELECT * FROM WORKER;
```

	Worker_id	FirstName	middleName	lastname	POST	salary
▶	1001	JAYRAM	PRAKASH	YADAV	Manager	25000
	1002	RAJA	RAOJI	YADAV	Helper	22000
	1003	PRAKASH	SADASHIV	MORE	Distributor	20000
	1004	KIRAN	SADASHIV	MORE	Distributor	20000

SELECT * FROM CUSTOMER;

[illegible]

12: SELECT QUERY WITH CONDITION –

Details of all worker working as distributor.

SELECT * FROM WORKER WHERE POST = "DISTRIBUTOR";

Worker_id	FirstName	middlename	lastname	POST	salary
1003	PRAKASH	SADASHIV	MORE	Distributor	20000
1004	KIRAN	SADASHIV	MORE	Distributor	20000
NULL	NULL	NULL	NULL	NULL	NULL

Details of all customer who living in sector 3 and ordering cow milk.

SELECT * FROM CUSTOMER WHERE CUST_DIVISION = "SECTOR 3" AND PRODUCT_ID = 1;

Cust_id	cust_fname	cust_mname	cust_lname	Cust_addline1	Cust_addline2	Cust_addline3	Pincode	Cust_division	Cust_order	product_id	Distributor_id
6	Jyoti	S	Jagdale	room no 02	Sai Darshan Apt B wing	Suresh Medical	NULL	Sector 3	2	1	1004
7	Kiran	J	Mhatre	room no 14	Sai Darshan Apt B wing	Suresh Medical	NULL	Sector 3	2	1	1004
8	Baji	Shambhu	Rhosale	room no 22	Sai Darshan Apt B wing	Suresh Medical	NULL	Sector 3	3	1	1004
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Details of all customer who living in sector 3 and ordering Buffalo milk.

SELECT * FROM CUSTOMER WHERE CUST_DIVISION = "SECTOR 3" AND PRODUCT_ID = 2;

Cust_id	cust_fname	cust_mname	cust_lname	Cust_addline1	Cust_addline2	Cust_addline3	Pincode	Cust_division	Cust_order	product_id	Distributor_id
5	Raju	J	R	room no 01	Sai Darshan Apt A wing	Suresh Medical	NULL	Sector 3	2	2	1004
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13: SUB QUERY –

Second highest salary from worker.

SELECT MAX(SALARY) AS SECONDHIGHESTSALARY FROM WORKER WHERE SALARY < (SELECT MAX(SALARY) FROM WORKER);

SECONDHIGHESTSALARY
22000

Second minimum salary from worker.

SELECT MIN(SALARY) AS SECONDMINSALARY FROM WORKER WHERE SALARY > (SELECT MIN(SALARY) FROM WORKER);

SECONDMINSALARY
22000

Second minimum order from customer.

SELECT MIN(CUST_ORDER) FROM CUSTOMER WHERE CUST_ORDER > (SELECT MIN(CUST_ORDER) FROM CUSTOMER);

MIN(CUST_ORDER)
2

14: JOINT QUERY –

LEFT JOIN

SELECT WORKER.WORKER_ID, CUSTOMER.DISTRIBUTOR_ID FROM WORKER LEFT JOIN CUSTOMER
ON WORKER.WORKER_ID = CUSTOMER.DISTRIBUTOR_ID;

WORKER_ID	DISTRIBUTOR_ID
1001	
1002	
1003	1003
1003	1003
1003	1003
1003	1003
1004	1004
1004	1004
1004	1004
1004	1004

RIGHT JOIN

SELECT WORKER.WORKER_ID, CUSTOMER.DISTRIBUTOR_ID FROM WORKER RIGHT JOIN
CUSTOMER ON WORKER.WORKER_ID = CUSTOMER.DISTRIBUTOR_ID;

WORKER_ID	DISTRIBUTOR_ID
1003	1003
1003	1003
1003	1003
1003	1003
1004	1004
1004	1004
1004	1004
1004	1004

UNION ALL

SELECT PRODUCTS.PRODUCT_ID AS PRD_ID, CUSTOMER.PRODUCT_ID FROM PRODUCTS RIGHT
JOIN CUSTOMER ON PRODUCTS.PRODUCT_ID = CUSTOMER.PRODUCT_ID;

PRD_ID	PRODUCT_ID
1	1
1	1
1	1
1	1
1	1
1	1
2	2
2	2

Conclusion:

The “Milk Management System” developed using SQL offers a robust solution for efficiently managing various aspects of worker details, distribution, and sales. By leveraging the power of SQL databases, the system ensures data integrity, scalability, and ease of access for users at different levels within the organization.

The system encompasses several key features, including:

➤ **Worker Management:**

Recording worker names, post and salary details ect.

➤ **Supplier Management :**

Recording supplier details, delivery schedules, and maintaining a database of reliable suppliers.

➤ **Customer Management :**

Managing customer accounts, orders, and delivery preferences to ensure timely and accurate deliveries.

➤ **Sales and Billing Management :**

Generating invoices, tracking sales transactions, and managing billing information for both wholesale and retail customers.

By utilizing SQL queries and database operations, the system efficiently handles complex data manipulations, joins, and aggregations required for seamless operations. The use of indexes, constraints, and normalization techniques ensures optimal database performance and data consistency.

In summary, the milk management system implemented in SQL not only streamlines day-to-day operations but also lays the foundation for innovation and growth in the dairy industry, empowering businesses to thrive in a competitive market landscape.

