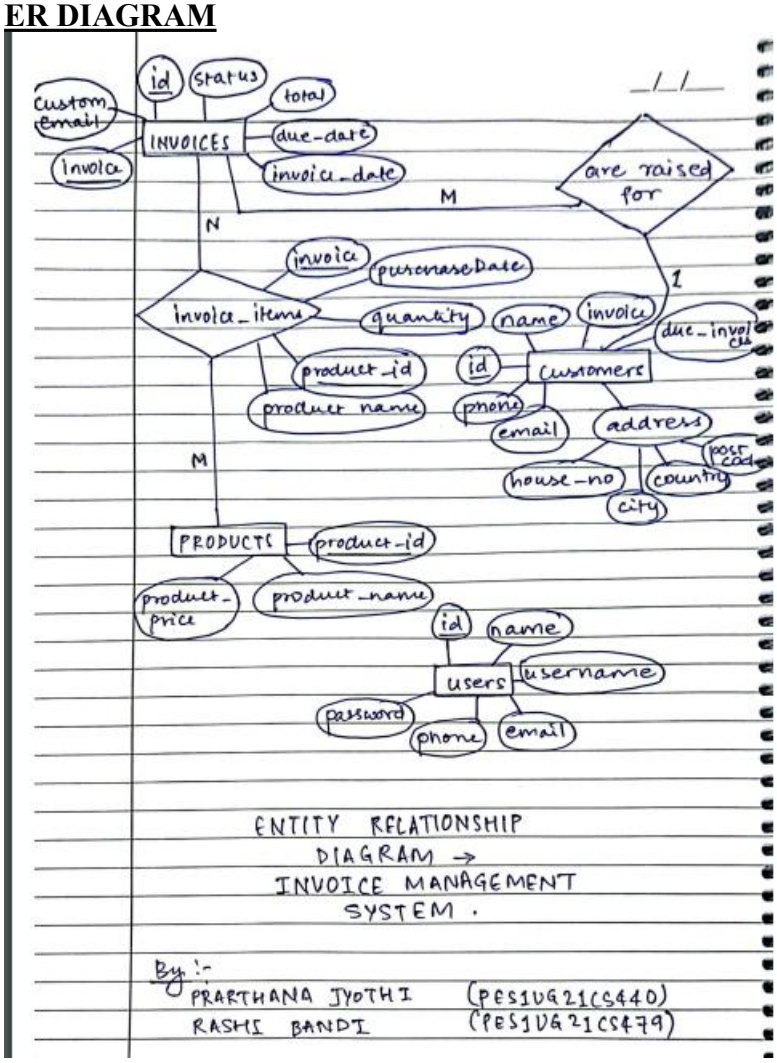
# DBMS MINIPROJECT

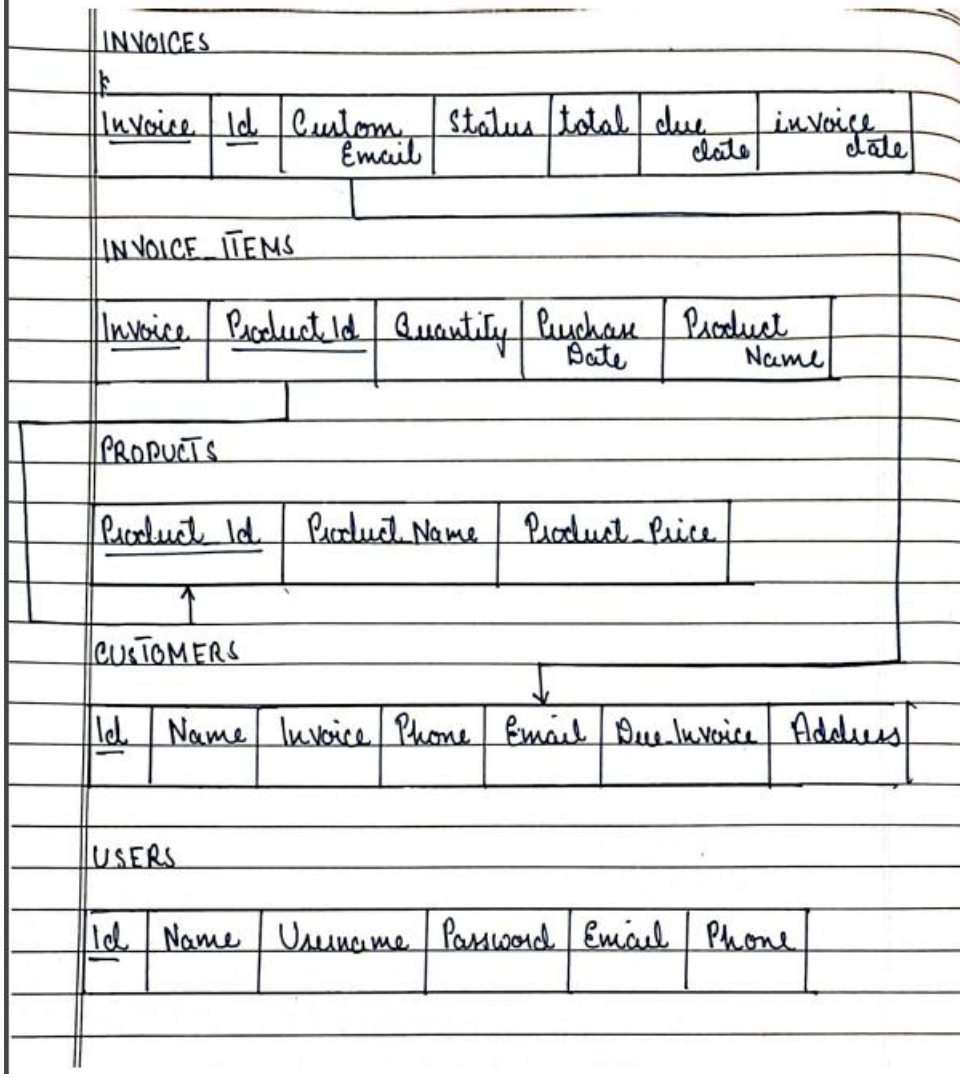
**Prarthana Jyothi- PES1UG21C440**
**Rashi Bnadi- PES1UG21CS479**

---

**ER DIAGRAM**



**RELATIONAL SCHEMA**



## CREATE COMMAND

We used the create command to create all the tables I.e Customers, Invoices, Products ,Invoice_items and Users.

```
CREATE TABLE `customers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `invoice` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `house_no` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `country` varchar(255) NOT NULL,
  `postcode` varchar(255) NOT NULL,
  `phone` varchar(20) NOT NULL, -- Changed from int(20) to varchar(20)
  `address` varchar(255),
  `due_invoices` INT DEFAULT 0,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `invoices` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `invoice` varchar(255) NOT NULL UNIQUE,
  `custom_email` varchar(255) NOT NULL,
  `invoice_date` date NOT NULL,
  `invoice_due_date` date NOT NULL,
  `total` decimal(10,0) NOT NULL,
  `status` varchar(255) NOT NULL default 'open',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `products` (
  `product_id` int(11) NOT NULL,
  `product_name` text NOT NULL,
  `product_price` varchar(255) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `username` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `phone` varchar(100) NOT NULL,
  `password` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `invoice_items`(
    `invoice` varchar(255) NOT NULL,
    `product_id` int(11) NOT NULL,
    `product_name` varchar(255) NOT NULL,
    `qty` int NOT NULL,
    `purchaseDate` date
);
```

**READ COMMAND**

We used the Select command in many places in our project especially in our dashboard page.There are few of the commands we used.

```
mysql> SELECT SUM(total) AS total_amount_due FROM invoices WHERE status = 'open';
+------------------+
| total_amount_due |
+------------------+
|             3045 |
+------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT SUM(total) AS total_amount_received FROM invoice
s WHERE status = 'paid';
+------------------------+
| total_amount_received |
+------------------------+
|                    858 |
+------------------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS total_invoices FROM invoices;;
+-----------------+
| total_invoices |
+-----------------+
|              13 |
+-----------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT COUNT(*) AS total_invoices FROM invoices;
+----------------+
| total_invoices |
+----------------+
|             13 |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS total_invoices_due FROM invoices WHE
RE status = 'open';
+--------------------+
| total_invoices_due |
+--------------------+
|                 10 |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(DISTINCT c.id) AS customers_with_pending_i
nvoices FROM customers c INNER JOIN invoices i ON c.email = i.
custom_email WHERE i.status = 'open';
+--------------------------------+
| customers_with_pending_invoices |
+--------------------------------+
|                              8 |
+--------------------------------+
1 row in set (0.00 sec)
```



**UPDATE COMMAND**

We used the update command to update the status of our invoice from open to paid.

```
# Update the status in the database
update_query = "UPDATE invoices SET status = %s WHERE id = %s"
cursor.execute(update_query, (new_status, invoice_id))
connection.commit()
connection.close()
```

**DELETE COMMAND**

Delete command is used to delete the products if and when the user whats to dete it in the manage products page.

```
# Delete selected products from the database
delete_query = f"DELETE FROM products WHERE product_id IN ({product_ids_str})"
cursor.execute(delete_query)
connection.commit()
```

**NESTED QUERY**

We used the nested query in our dashboard page to print the most no.of products bought in a month.

```
mysql> SELECT product_name, COALESCE(SUM(qty), 0) AS total_quantity_sold
    ->                              FROM invoice_items
    ->                              WHERE (YEAR(purchaseDate), MONTH(purchaseDate)) =
    ->                                  (SELECT YEAR(CURDATE()), MONTH(CURDATE()))
    ->                              GROUP BY product_name
    ->                              ORDER BY total_quantity_sold DESC
    ->                              LIMIT 1;
+---------------+---------------------+
| product_name  | total_quantity_sold |
+---------------+---------------------+
| Product Three |                  19 |
+---------------+---------------------+
1 row in set (0.00 sec)
```

## TRIGGER

```
DELIMITER //

CREATE TRIGGER create_address BEFORE INSERT ON customers
FOR EACH ROW
BEGIN
    SET NEW.address = CONCAT_WS(' ', NEW.house_no, NEW.city, NEW.country, NEW.postcode);
END;
//
DELIMITER ;
```

```
DELIMITER //
CREATE TRIGGER update_due_invoices_on_status_change
AFTER UPDATE ON invoices
FOR EACH ROW
BEGIN
    IF NEW.status <> OLD.status THEN
        CALL update_due_invoices(NEW.custom_email);
    END IF;
END//
DELIMITER ;
```

```
DELIMITER //
CREATE TRIGGER update_due_invoices_on_insert
AFTER INSERT ON invoices
FOR EACH ROW
BEGIN
    IF NEW.status = 'open' THEN
        CALL update_due_invoices(NEW.custom_email);
    END IF;
END//
DELIMITER ;
```

**PROCEDURE**

```
DELIMITER //
CREATE PROCEDURE update_due_invoices(IN customer_email VARCHAR(255))
BEGIN
    DECLARE num_due_invoices INT;
    
    SELECT COUNT(*) INTO num_due_invoices
    FROM invoices
    WHERE custom_email = customer_email AND status = 'open';

    UPDATE customers
    SET due_invoices = num_due_invoices
    WHERE email = customer_email;
END//
DELIMITER ;
```