



PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering

B.TECH.
(ECE) IV
SEMESTER
UE20CS301 – Database Management Systems (Minors)
Mini-Project Report
on
Electronics Components Marketplace

SUBMITTED BY

Name	SRN
Gautham Varma K	PES1UG21EC102
Nishchal AV	PES1UG21EC169
Nikhil R	PES1UG21EC164
Mrinal Pradeep	PES1UG21EC152

January – May
2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BENGALURU – 560085, KARNATAKA, INDIA

Electronic Components Marketplace ⚡

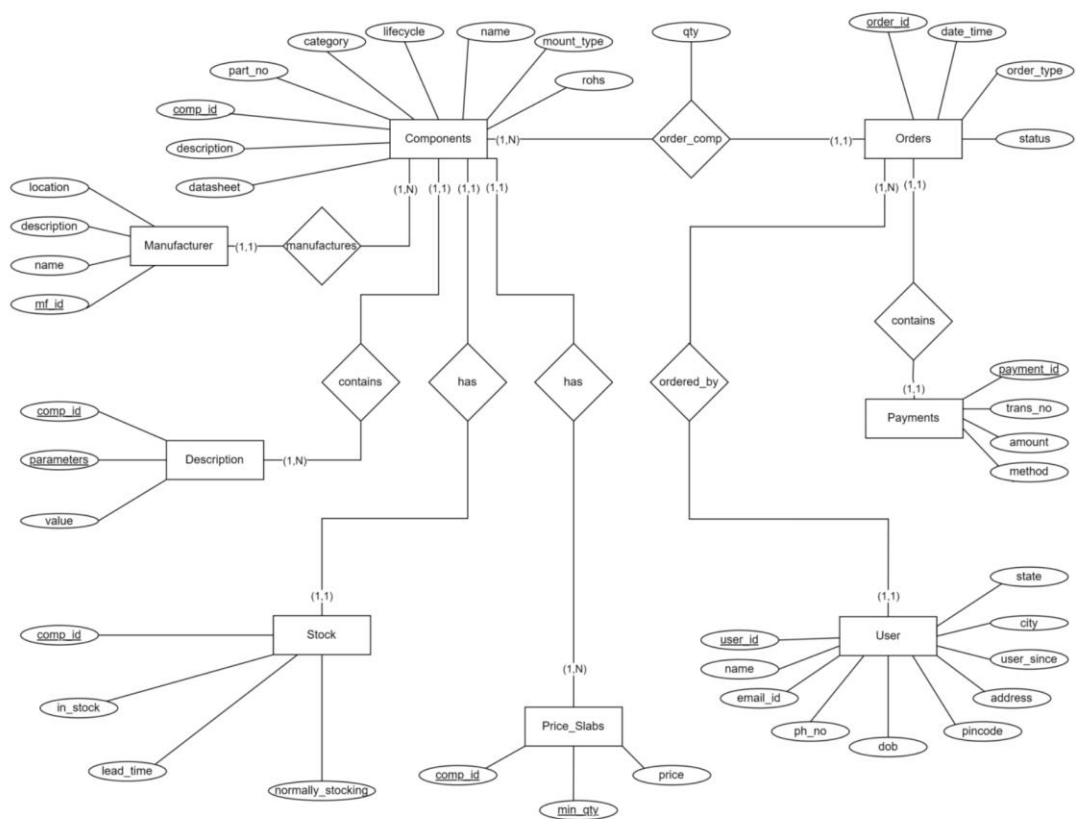
Mini-Project created for the partial fulfilment of the course [UE20CS301] DBMS, CSE Minors, PES University.

By:

SRN	Name
PES1UG21EC102	Gautham Varma K
PES1UG21EC169	Nishchal A V
PES1UG21EC164	Nikhil R
PES1UG21EC152	Mrinal Pradeep

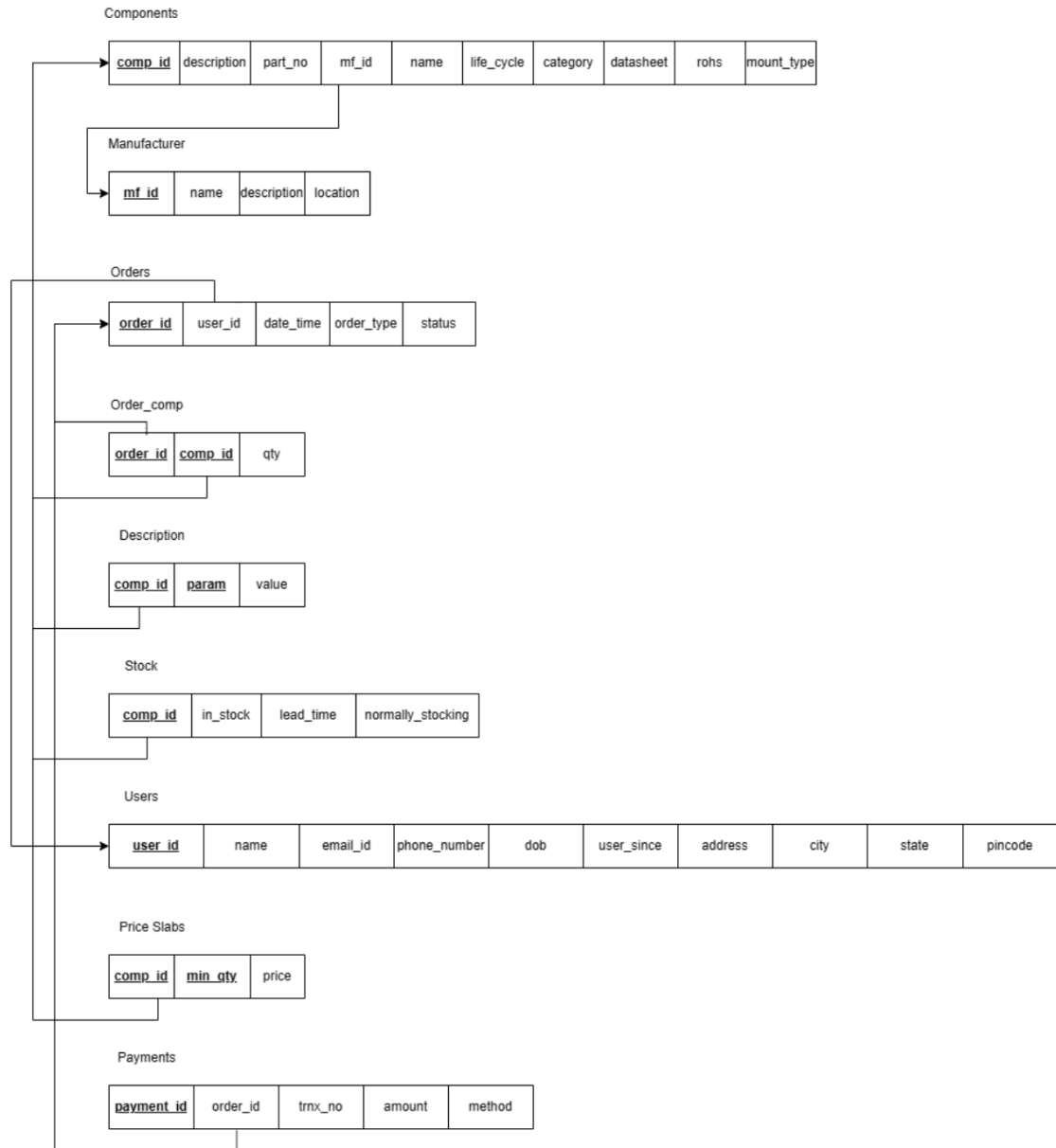
ER Diagram

This is the ER Diagram of the database.



Relational Schema

This is the Relational Schema of the database.



Database Overview

This is the overview of the database.

```
CREATE DATABASE electronics_store;
```

```
USE electronics_store;
```

```
CREATE TABLE users(  
    user_id VARCHAR(50) PRIMARY KEY,  
    name VARCHAR(50),  
    email_id VARCHAR(50) CHECK (email_id LIKE '%@%.%'),  
    phone_number VARCHAR(50) CHECK (LENGTH(phone_number) = 10),  
    dob DATE,  
    user_since DATE,  
    address VARCHAR(100),  
    city VARCHAR(50),  
    state VARCHAR(50),  
    pincode INT  
);
```

```
CREATE TABLE manufacturer(  
    mf_id VARCHAR(50) PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    description VARCHAR(100),  
    location VARCHAR(50)  
);
```

```
CREATE TABLE components(  
    comp_id INT PRIMARY KEY,  
    description VARCHAR(50),  
    part_no VARCHAR(50) NOT NULL,  
    mf_id VARCHAR(50),  
    name VARCHAR(50) NOT NULL,  
    lifecycle VARCHAR(50),  
    category VARCHAR(50),  
    datasheet VARCHAR(100),  
    rohs VARCHAR(50),  
    mount_type VARCHAR(50),  
    FOREIGN KEY(mf_id) REFERENCES manufacturer(mf_id)  
);
```

```
CREATE TABLE description(  
    comp_id INT,  
    param VARCHAR(50),  
    value VARCHAR(50),  
    FOREIGN KEY(comp_id) REFERENCES components(comp_id),  
    PRIMARY KEY(comp_id, param)  
);
```

```
CREATE TABLE stock(  
    comp_id INT,  
    param VARCHAR(50),  
    value VARCHAR(50),  
    FOREIGN KEY(comp_id) REFERENCES components(comp_id),  
    PRIMARY KEY(comp_id, param)
```

```

    comp_id INT,
    in_stock INT,
    lead_time INT,
    normally_stocking VARCHAR(5),
    FOREIGN KEY(comp_id) REFERENCES components(comp_id),
    PRIMARY KEY(comp_id)
);

CREATE TABLE price_slabs(
    comp_id INT,
    min_qty INT,
    price DECIMAL(6,2) NOT NULL,
    FOREIGN KEY(comp_id) REFERENCES components(comp_id),
    PRIMARY KEY(comp_id, min_qty)
);

CREATE TABLE orders(
    order_id INT PRIMARY KEY,
    user_id VARCHAR(50),
    date_time DATETIME,
    order_type VARCHAR(50),
    status VARCHAR(50),
    FOREIGN KEY(user_id) REFERENCES users(user_id)
);

CREATE TABLE order_comp(
    order_id INT,
    comp_id INT,
    qty INT,
    FOREIGN KEY(order_id) REFERENCES orders(order_id),
    FOREIGN KEY(comp_id) REFERENCES components(comp_id),
    PRIMARY KEY(order_id, comp_id)
);

CREATE TABLE payments(
    payment_id INT PRIMARY KEY,
    order_id INT,
    trnx_no VARCHAR(50),
    amount INT,
    method VARCHAR(50),
    FOREIGN KEY(order_id) REFERENCES orders(order_id)
);

CREATE VIEW user_age AS
SELECT user_id, name, email_id, phone_number, dob, TIMESTAMPDIFF(YEAR,
dob, CURDATE()) AS age, user_since, address, city, state, pincode
FROM users;

```

Users

	user_id	name	email_id	phone_number	dob	user_since	address	city	state	pincode
0	a1	Kaushik	abc@gmail.com	8756438291	2004-02-12	2012-12-12	Ulsoor	Bangalore	Karnataka	560,093
1	a2	Monish	eee@gmail.com	9956438291	2003-10-10	2023-12-01	KumbleCircle	Bangalore	Karnataka	560,093
2	a3	Nitin	dcf@gmail.com	9956411111	2000-11-10	2023-12-01	RRNagar	Bangalore	Karnataka	560,012
3	a4	Sajith	slv@gmail.com	6877798222	2003-11-05	2009-12-12	Indiranagar	Bangalore	Karnataka	560,093
4	b7	Rajesh	acd@gmail.com	8756438671	2003-02-12	2021-07-12	Churchstreet	Bangalore	Karnataka	560,093

Manufacturer

	mf_id	name	description	location
0	1	Lite-On Inc.	Building LEDS since 1983, leading producer	Mumbai
1	2	Rutherford Tech	Building LEDS since 1983, leading producer	Kolkata
2	3	TATA Inc.	The most trusted Manufacturers	Mumbai
3	4	MasalaDosaResistors	Namma Bengaluru special	Bangalore

Components

	comp_id	description	part_no	mf_id	name	lifecycle	category	datash	rohs	mount_type
0	16,014,451	LED ORANGE CLEAR SMD	LTST-C191KFKT	1	Orange SMD LED	NFNPD	LEDS	https://	Compliant	Surface Mount
1	16,014,452	LED GREEN CLEAR SMD	LTST-C191KGKT	1	Green SMD LED	NFNPD	LEDS	https://	Compliant	Surface Mount
2	16,014,453	LED RED CLEAR SMD	LTST-C191KRKT	1	Red SMD LED	InProduction	LEDS	https://	Compliant	Surface Mount
3	16,014,454	LED YELLOW CLEAR SMD	LTST-C191KSKT	2	Yellow SMD LED	InProduction	LEDS	https://	Compliant	Surface Mount
4	16,014,455	RESISTOR	RTST-S250KGKT	3	2.5k SMRA Resistor	InProduction	LEDS	https://	Compliant	Surface Mount Right Angle
5	16,014,456	RESISTOR	RTST-S270KGKT	3	220 SMD Resistor	InProduction	Resistors	https://	Compliant	Surface Mount
6	16,014,457	RESISTOR	RTST-S271CGKT	4	45 SMD Resistor	InProduction	Resistors	https://	Compliant	Surface Mount

Description

	comp_id	param	value
0	16,014,451	Voltage	3 Volts
1	16,014,452	Voltage	3.2 Volts
2	16,014,453	Voltage	3 Volts
3	16,014,454	Voltage	1.9 Volts
4	16,014,455	Resistance	2500 Ohms
5	16,014,456	Resistance	220 Ohms
6	16,014,457	Resistance	45 Ohms

Stock

	comp_id	in_stock	lead_time	normally_stocking
0	16,014,451	56,000	None	yes
1	16,014,452	19	9	yes
2	16,014,453	7	15	no
3	16,014,454	16	0	yes
4	16,014,455	26,000	9	yes
5	16,014,456	96,754	9	no
6	16,014,457	402	2	yes

Price Slabs

	comp_id	min_qty	price
0	16,014,451	50	2.50
1	16,014,451	500	2.30
2	16,014,451	5,000	2.00
3	16,014,452	10	2.50
4	16,014,452	100	2.30
5	16,014,452	1,000	2.00
6	16,014,453	15	2.00
7	16,014,453	150	1.80
8	16,014,453	1,500	1.50
9	16,014,454	40	1.20

Orders

	order_id	user_id	date_time	order_type	status
0	256	a1	2023-04-27 00:00:00	Regular	ToBeShipped
1	257	a2	2023-04-01 00:00:00	Prime	ToBeShipped
2	258	a3	2020-04-01 00:00:00	Prime	Delivered
3	259	a4	2023-04-14 00:00:00	BackOrder	Shipped
4	722	b7	2012-04-14 00:00:00	Regular	Missing

Order Components

	order_id	comp_id	qty
0	256	16,014,451	450
1	256	16,014,452	550
2	256	16,014,453	600
3	256	16,014,457	650
4	257	16,014,451	12,500
5	257	16,014,452	600
6	258	16,014,453	9,000
7	258	16,014,457	125
8	259	16,014,454	200
9	722	16,014,457	1,200

Payments

	payment_id	order_id	trnx_no	amount	method
0	130	256	5678AB1002	12,000	Cash
1	131	257	5678AB1042	14,000	Cash
2	132	258	5600AB1342	25,000	Cash
3	133	259	5678AB1152	567	UPI
4	134	722	4978AB1152	20,000	Cheque

User Age [View]

	user_id	name	email_id	phone_number	dob	age	user_since	address	city	state	pincode
0	a1	Kaushik	abc@gmail.com	8756438291	2004-02-12	19	2012-12-12	Ulsoor	Bangalore	Karnataka	560,093
1	a2	Monish	eee@gmail.com	9956438291	2003-10-10	19	2023-12-01	KumbleCircle	Bangalore	Karnataka	560,093
2	a3	Nitin	dcf@gmail.com	9956411111	2000-11-10	22	2023-12-01	RRNagar	Bangalore	Karnataka	560,012
3	a4	Sajith	slv@gmail.com	6877798222	2003-11-05	19	2009-12-12	Indiranagar	Bangalore	Karnataka	560,093
4	b7	Rajesh	acd@gmail.com	8756438671	2003-02-12	20	2021-07-12	Churchstreet	Bangalore	Karnataka	560,093

Components CRUD UI

This is the CRUD UI for the components table.

Table of all components

`SELECT * FROM components;`

	comp_id	description	part_no	mf_id	name	lifecycle	category	datash	rohs	mount_type
0	16,014,451	LED ORANGE CLEAR SMD	LTST-C191KFKT	1	Orange SMD LED	NFNPD	LEDs	https://	Compliant	Surface Mount
1	16,014,452	LED GREEN CLEAR SMD	LTST-C191KGKT	1	Green SMD LED	NFNPD	LEDs	https://	Compliant	Surface Mount
2	16,014,453	LED RED CLEAR SMD	LTST-C191KRKT	1	Red SMD LED	InProduction	LEDs	https://	Compliant	Surface Mount
3	16,014,454	LED YELLOW CLEAR SMD	LTST-C191KSKT	2	Yellow SMD LED	InProduction	LEDs	https://	Compliant	Surface Mount
4	16,014,455	RESISTOR	RTST-S250KGKT	3	2.5k SMRA Resistor	InProduction	LEDs	https://	Compliant	Surface Mount Right Angle
5	16,014,456	RESISTOR	RTST-S270KGKT	3	220 SMD Resistor	InProduction	Resistors	https://	Compliant	Surface Mount
6	16,014,457	RESISTOR	RTST-S271CGKT	4	45 SMD Resistor	InProduction	Resistors	https://	Compliant	Surface Mount

Add a new component

Component ID

0.00

-

+

Description

Part Number

Manufacturer ID

Name

Lifecycle

Category

Datasheet

RoHS

Mount Type

Add

Update an existing component

Component ID

0.00- +

Description

Part Number

Manufacturer ID

Name

Lifecycle

Category

Datasheet

RoHS

Mount Type

Update

Delete an existing component

Component ID

0.00- +

Delete

Search for a component

Component ID

0.00- +

Search

Gautham's Queries

This is the Queries page by Gautham Varma K [PES1UG21EC102].

Query 1

Display the top 20 orders this year sorted by price.

```
SELECT orders.order_id, user_id, date_time, order_type, pm.amount
FROM orders, payments AS pm
WHERE order_id = pm.order_id AND date_time LIKE '2023%'
ORDER BY pm.amount DESC
LIMIT 20;
```

	order_id	user_id	datetime	order_type	price
0	257	a2	2023-04-01 00:00:00	Prime	14,000
1	256	a1	2023-04-27 00:00:00	Regular	12,000
2	259	a4	2023-04-14 00:00:00	BackOrder	567

Query 2

Find all resistors that are of 220 Ohms that are in Production

```
SELECT *
FROM components
WHERE (comp_id IN (SELECT comp_id FROM description WHERE param =
'Resistance' AND value = '220 Ohms')) AND lifecycle = 'InProduction';
```

	comp_id	description	part_no	mf_id	name	lifecycle	category	datasheet	rohs	mount_type
0	16,014,456	RESISTOR	RTST-S270KGKT	3	220 SMD Resistor	InProduction	Resistors	https://optoelectroi	Compliant	Surface Mount

Query 3

List up to 50 components that have less than 50 pieces in stock but have a lead time of greater than 8 weeks.

```
SELECT components.comp_id, components.name, components.part_no,
components.mf_id, components.lifecycle, components.category,
components.datasheet, components.rohs, components.mount_type,
stock.in_stock, stock.location, stock.lead_time
FROM components,stock
WHERE components.comp_id = stock.comp_id AND stock.in_stock < 50 AND
stock.lead_time > 8
LIMIT 50;
```

	comp_id	name	part_no	mf_id	lifecycle	category	datasheet	rohs	mount_type	quantity	lead_time	normally_stocking
0	16,014,452	Green SMD LED	LTST-C191KGKT	1	NFNPD	LEDS	https://	Compliant	Surface Mount	19	9	yes
1	16,014,453	Red SMD LED	LTST-C191KRKT	1	InProduction	LEDS	https://	Compliant	Surface Mount	7	15	no

Query 4

List the top 10 products sold by a given manufacturer based on the total number of pieces sold.

```
SELECT components.comp_id, components.name, total_pieces_sold
FROM components, (
    SELECT comp_id AS cmpid, SUM(order_comp.qty) AS total_pieces_sold
    FROM order_comp
    GROUP BY comp_id
) AS comp_sold
WHERE components.comp_id = comp_sold.cmpid AND components.mf_id = '1'
ORDER BY total_pieces_sold DESC
LIMIT 10;
```

	Component ID	Name	Total Pieces Sold
0	16,014,451	Orange SMD LED	12950.
1	16,014,453	Red SMD LED	9600.
2	16,014,452	Green SMD LED	1150.

Query 5

Only list components that are RoHS compliant sold by a certain manufacturer

```
SELECT *
FROM components
WHERE components.rohs = 'Compliant' AND components.mf_id = '1';
```

	comp_id	description	part_no	mf_id	name	lifecycle	category	datash	rohs	mount_type
0	16,014,451	LED ORANGE CLEAR SMD	LTST-C191KFKT	1	Orange SMD LED	NFNPD	LEDS	https://	Compliant	Surface Mount
1	16,014,452	LED GREEN CLEAR SMD	LTST-C191KGKT	1	Green SMD LED	NFNPD	LEDS	https://	Compliant	Surface Mount
2	16,014,453	LED RED CLEAR SMD	LTST-C191KRKT	1	Red SMD LED	InProduction	LEDS	https://	Compliant	Surface Mount

Query 6

List orders whose status is Not shipped if placed before a week

```
SELECT *
FROM orders
WHERE status = 'ToBeShipped' AND date_time < DATE(DATE_SUB(NOW(),
INTERVAL 1 WEEK));
```

	order_id	user_id	date_time	order_type	status
0	257	a2	2023-04-01 00:00:00	Prime	ToBeShipped

Query 7

Find the Most commonly used mode of payment

```
SELECT method AS payment_mode, COUNT(method)
FROM payments
GROUP BY payment_mode
ORDER BY COUNT(payment_mode) DESC
LIMIT 1;
```

	payment_mode	count
0	Cash	3

Query 8

Select resistors of 45 Ohms components that are 'InProduction' and have in_stock of greater than 400

```
SELECT components.*, stock.in_stock, stock.lead_time
FROM components, stock, description
WHERE components.comp_id = stock.comp_id AND components.comp_id =
description.comp_id AND description.param = 'Resistance' AND
description.value = '45 Ohms' AND components.lifecycle = 'InProduction'
AND stock.in_Stock > 400;
```

	comp_id	description	part_no	mf_id	lifecycle	name	category	datasheet	rohs	mount_type	quantity	lead_time
0	16,014,457	RESISTOR	RTST-S271CGKT	4	45 SMD Resistor	InProduction	Resistors	https://o	Compliant	Surface Mount	402	2

Nishchal's Queries

This is the Queries page by Nishchal A V [PES1UG21EC169].

Query 1

List the components that are limited in stock.

```
SELECT stock.comp_id,manufacturer.name AS mf_name,components.name AS comp_name,components.datasheet,components.part_no,stock.normally_stocking,stock.in_stock,price_slabs.price,min_qty
FROM stock
INNER JOIN components ON components.comp_id = stock.comp_id
INNER JOIN manufacturer ON manufacturer.mf_id = components.mf_id
JOIN price_slabs ON price_slabs.comp_id = components.comp_id AND stock.in_stock < 100000 AND (SELECT(stock.normally_stocking = 'no')) AND min_qty <= 100;
```

	comp_id	mf_name	comp_name	datasheet	part_no	normally_stocking	in_stock	price	min_qty
0	16,014,453	Lite-On Inc.	Red SMD LED	https://optoele	LTST-C191KRKT	no	7	2.00	15
1	16,014,456	TATA Inc.	220 SMD Resistor	https://optoele	RTST-S270KGKT	no	96,754	2.40	30

Query 2

Tabulate orders with respect to user order type.

```
SELECT users.user_id,orders.order_id,orders.date_time,orders.order_type
FROM users,orders
WHERE orders.user_id = users.user_id AND orders.date_time < '2022-07-01 00:00:00' AND order_type = 'Prime';
```

	user_id	order_id	date_time	order_type
0	a3	258	2020-04-01 00:00:00	Prime

Query 3

Give the best price slabs with respect to components

```
SELECT components.comp_id,price_slabs.min_qty,price_slabs.price,price_slabs.price * price_slabs.min_qty total_price,components.name
FROM price_slabs
JOIN components ON price_slabs.comp_id = components.comp_id
WHERE price <= 3 AND min_qty <= 100;
```

	comp_id	min_qty	price	total_price	name
0	16,014,451	50	2.50	125.00	Orange SMD LED
1	16,014,452	10	2.50	25.00	Green SMD LED
2	16,014,452	100	2.30	230.00	Green SMD LED
3	16,014,453	15	2.00	30.00	Red SMD LED
4	16,014,454	40	1.20	48.00	Yellow SMD LED
5	16,014,455	32	1.70	54.40	2.5k SMRA Resistor
6	16,014,456	30	2.40	72.00	220 SMD Resistor

Query 4

Check how many manufacturers are selling surface mount components with lifecycle of InProduction and the components is resistor.

SELECT

components.mf_id,manufacturer.name,manufacturer.location,components.lifecycle,components.category

FROM manufacturer

JOIN components **ON** components.mf_id = manufacturer.mf_id **AND**

(**SELECT**(mount_type='Surface Mount')) **AND**

(**SELECT**(lifecycle='InProduction')) **AND** components.category = 'Resistors';

	mf_id	name	location	lifecycle	category
0	3	TATA Inc.	Mumbai	InProduction	Resistors
1	4	MasalaDosaResistors	Bangalore	InProduction	Resistors

Query 5

Track users who are paying through cash

SELECT

payments.method,orders.order_id,orders.user_id,users.name,phone_number, address,city,state,pincode,payments.trnx_no,payments.amount

FROM payments,orders,users

WHERE orders.user_id = users.user_id **AND** orders.order_id =

payments.order_id **AND** payments.method = 'cash';

	method	order_id	user_id	name	phone_number	address	city	state	pincode	trnx_no	amount
0	Cash	256	a1	Kaushik	8756438291	Ulsoor	Bangalore	Karnataka	560,093	5678AB1002	12,000
1	Cash	257	a2	Monish	9956438291	KumbleCircle	Bangalore	Karnataka	560,093	5678AB1042	14,000
2	Cash	258	a3	Nitin	9956411111	RRNagar	Bangalore	Karnataka	560,012	5600AB1342	25,000

Nikhil's Queries

This is the Queries page by Nikhil R [PES1UG21EC164].

Query 1

Select all components whose lead time is minimum from table Stock

```
SELECT *  
FROM stock  
WHERE lead_time < 1;
```

	comp_id	in_stock	lead_time	normally_stocking
0	16,014,454	16	0	yes

Query 2

Retrieve all details from the table price_slabs where the price is unusually high

```
SELECT *  
FROM price_slabs  
WHERE price_slabs.price > (SELECT AVG(price) FROM price_slabs);
```

	comp_id	min_qty	price
0	16,014,451	50	2.50
1	16,014,452	10	2.50
2	16,014,457	124	7.00
3	16,014,457	1,240	6.50
4	16,014,457	12,400	6.00

Query 3

Display those locations which have more than one manufacturer stationed.

```
SELECT location, COUNT(*) AS num  
FROM manufacturer  
GROUP BY(location)  
HAVING COUNT(*) > 1;
```

	location	num
0	Mumbai	2

Query 4

Display the user_id, name and phone number of customers whose order status is Missing

```
SELECT users.user_id, users.name, users.phone_number, orders.status
FROM users
LEFT JOIN orders ON users.user_id=orders.user_id AND
orders.status='Missing';
```

	user_id	name	phone_number	status
0	b7	Rajesh	8756438671	Missing
1	a1	Kaushik	8756438291	None
2	a2	Monish	9956438291	None
3	a3	Nitin	9956411111	None
4	a4	Sajith	6877798222	None

Query 5

Display the user id of those users who have made payments larger than 20000.

```
SELECT users.user_id, orders.order_id, payments.trnx_no,
payments.amount
FROM users
INNER JOIN orders ON users.user_id=orders.user_id
INNER JOIN payments ON payments.order_id=orders.order_id
WHERE payments.amount>20000;
```

	user_id	order_id	trnx_no	amount
0	a3	258	5600AB1342	25,000

Mrinal's Queries

This is the Queries page by Mrinal [PES1UG21EC152].

Query 1

Show the number of LEDs components that are in production

```
SELECT COUNT(category) AS count
FROM components
WHERE category='LEDS' AND lifecycle='InProduction';
```

Count
3

Query 2

Select the cheapest component(using comp_ID)

```
SELECT comp_id
FROM price_slabs
WHERE price=(SELECT MIN(price) FROM price_slabs);
```

Count
16,014,454

Query 3

Find the name of a newer user, whose order has to be shipped and find their mode of payment

```
SELECT users.name,payments.method
FROM users,orders,payments
WHERE users.user_since = (SELECT max(user_since) FROM users) AND
users.user_ID = orders.user_ID AND orders.order_ID = payments.order_ID
AND orders.status = 'tobeshipped';
```

Name	Method
Monish	Cash

Query 4

Select users who has been a user for a long time and find their most use method of payments

```
SELECT users.user_ID,users.name,payments.method
FROM users,orders,payments
WHERE user_since=(SELECT MIN(user_since) FROM users) AND
users.user_ID=orders.user_ID AND orders.order_ID=payments.order_ID AND
payments.method=(SELECT payments.method FROM payments WHERE
payments.order_ID=orders.order_ID GROUP BY payments.method LIMIT 1);
```

User_ID	Name	Method
a4	Sajith	UPI

Query 5

Find the name of a user who has bought a component of mount_type="surface mount"

```
SELECT users.name
FROM users,components,orders,order_comp
WHERE users.user_ID=orders.user_ID AND
orders.order_ID=order_comp.order_ID AND
order_comp.comp_ID=components.comp_ID AND
components.mount_type="Surface Mount";
```

Name
Kaushik
Monish
Nitin
Sajith
Rajesh

Triggers

```
CREATE TABLE payments_backup(
    payment_id INT PRIMARY KEY,
    order_id INT,
    trnx_no VARCHAR(50),
    amount INT,
    method VARCHAR(50),
    deleted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY(order_id) REFERENCES orders(order_id)
);
```

```
DELIMITER $$
CREATE TRIGGER backup_payments
AFTER DELETE ON payments
FOR EACH ROW
BEGIN
    INSERT INTO payments_backup(payment_id, order_id, trnx_no, amount,
method)
    VALUES (OLD.payment_id, OLD.order_id, OLD.trnx_no, OLD.amount,
OLD.method);
END $$
DELIMITER ;
```

Triggering the Trigger

Payments Table

	payment_id	order_id	trnx_no	amount	method
0	130	256	5678AB1002	12,000	Cash
1	131	257	5678AB1042	14,000	Cash
2	132	258	5600AB1342	25,000	Cash
3	133	259	5678AB1152	567	UPI
4	134	722	4978AB1152	20,000	Cheque

Enter payment_id to delete

1

Delete

Payments Backup Table

	payment_id	order_id	trnx_no	amount	method	deleted_at

Functions

```
DELIMITER $$
CREATE FUNCTION is_prime_user(user_id VARCHAR(50)) RETURNS VARCHAR(5)
BEGIN
    DECLARE user_since DATE;
    DECLARE is_prime VARCHAR(5);

    SELECT users.user_since INTO user_since
    FROM users
    WHERE users.user_id = user_id;

    IF TIMESTAMPDIFF(YEAR, user_since, CURDATE()) > 4 THEN
        SET is_prime = 'True';
    ELSE
        SET is_prime = 'False';
    END IF;

    RETURN is_prime;
END;$$
DELIMITER ;
```

Is Prime User?

Users Table

	user_id	name	email_id	phone_number	dob	user_since	address	city	state	pincode
0	a1	Kaushik	abc@gmail.com	8756438291	2004-02-12	2012-12-12	Ulsoor	Bangalore	Karnataka	560,093
1	a2	Monish	eee@gmail.com	9956438291	2003-10-10	2023-12-01	KumbleCircle	Bangalore	Karnataka	560,093
2	a3	Nitin	dcf@gmail.com	9956411111	2000-11-10	2023-12-01	RRNagar	Bangalore	Karnataka	560,012
3	a4	Sajith	slv@gmail.com	6877798222	2003-11-05	2009-12-12	Indiranagar	Bangalore	Karnataka	560,093
4	b7	Rajesh	acd@gmail.com	8756438671	2003-02-12	2021-07-12	Churchstreet	Bangalore	Karnataka	560,093

Enter user_id to check

b7

Check

Is Prime User: False

Executed: SELECT is_prime_user('b7');

Procedures

```
DELIMITER $$
CREATE PROCEDURE restore_payment(IN payment_id1 INT)
BEGIN
    INSERT INTO payments(payment_id, order_id, trnx_no, amount, method)
    SELECT payment_id, order_id, trnx_no, amount, method
    FROM payments_backup
    WHERE payment_id = payment_id1;

    DELETE FROM payments_backup
    WHERE payment_id = payment_id1;
END $$
DELIMITER ;
```

Restoring the Payment

Payments Table

	payment_id	order_id	trnx_no	amount	method
0	130	256	5678AB1002	12,000	Cash
1	131	257	5678AB1042	14,000	Cash
2	132	258	5600AB1342	25,000	Cash
3	133	259	5678AB1152	567	UPI
4	134	722	4978AB1152	20,000	Cheque

Payments Backup Table

	payment_id	order_id	trnx_no	amount	method	deleted_at
empty						

Enter payment_id to restore

1

Restore

Bonus

Ever wondered how an order can be placed and processed??

1) Orders Table

Insert the order into the orders table.

```
INSERT INTO orders VALUES('300','a1','2023-05-03','Regular',  
'PaymentPending');  
SELECT * FROM orders WHERE order_id=300;
```

	order_id	customer_id	order_date	order_status	order_total
0	300	a1	2023-05-03 00:00:00	Regular	PaymentPending

2) Order Components Table

Insert the components into the order_comp table.

```
INSERT INTO order_comp VALUES('300','16014451','450');  
INSERT INTO order_comp VALUES('300','16014453','600');  
INSERT INTO order_comp VALUES('300','16014452','550');  
INSERT INTO order_comp VALUES('300','16014457','650');  
SELECT * FROM order_comp WHERE order_id=300;
```

	order_id	comp_id	quantity
0	300	16,014,451	450
1	300	16,014,452	550
2	300	16,014,453	600
3	300	16,014,457	650

3) Get a bill of the order

```
SELECT comp_id, comp_name, qty, price, amount  
FROM bill_view  
WHERE order_id = 300;
```

	comp_id	comp_name	qty	price	amount
0	16,014,451	Orange SMD LED	450	2.50	1125.00
1	16,014,452	Green SMD LED	550	2.30	1265.00
2	16,014,453	Red SMD LED	600	1.80	1080.00
3	16,014,457	45 SMD Resistor	650	7.00	4550.00

4) Get the total amount of the order

```
SELECT SUM(amount) FROM bill_view WHERE order_id = 300;
```

	total
0	8020.00

5) Payments Table

Insert the payment into the payments table.

```
INSERT INTO payments VALUES('300', '300', '123456789', '8020', 'COD');
UPDATE orders SET order_status='ToBeShipped' WHERE order_id=300;
SELECT * FROM orders WHERE order_id=300;
SELECT * FROM payments WHERE payment_id=300;
```

	order_id	customer_id	order_date	order_status	order_total
0	300	a1	2023-05-03 00:00:00	Regular	ToBeShipped

	payment_id	order_id	trnx_no	amount	method
0	300	300	123456789	2,250	COD

Now, how did we get the bill view?

```
CREATE VIEW bill_view AS
SELECT
    order_id,
    comp_id,
    NAME AS comp_name,
    qty,
    best_price(comp_id, qty) AS price,
    qty * best_price(comp_id, qty) AS amount
FROM order_comp
NATURAL JOIN components;
```

Now, how did we get the best price?

```
DELIMITER $$
CREATE FUNCTION best_price(comp_id INT, qty INT) RETURNS DECIMAL(6,2)
BEGIN
    DECLARE price DECIMAL(6,2);
    SELECT price_slabs.price INTO price
    FROM price_slabs
    WHERE price_slabs.comp_id=comp_id AND price_slabs.min_qty<qty
    ORDER BY price_slabs.min_qty DESC
    LIMIT 1;

    RETURN price;
```

```
END $$
DELIMITER ;
```

```

-----
|_  _| |      | |  \ \ / /
| | | |__  __  _ _ | |__ \ \ / /___  _  _
| | | | ' \ / _ ' | ' \ | / / \ \ / / _ \ | | |
| | | | | | ( _ | | | | <  | | ( _ ) | | _ |
\_ / | _ | | \ \_, | _ | | _ | \ \  \ \ / \_\_, |

```

```

_ _ _ _ _
| \ | | ( )
| . . | _ _ _ | / _ _ _ _ _
| | \ | | / _ ' | / _ ' | ' _ ' _ \
| | | | ( _ | | ( _ | | | | | |
\ _ | | _ / \ _ _ _ | \ _ _ _ _ | _ | | _ |

```