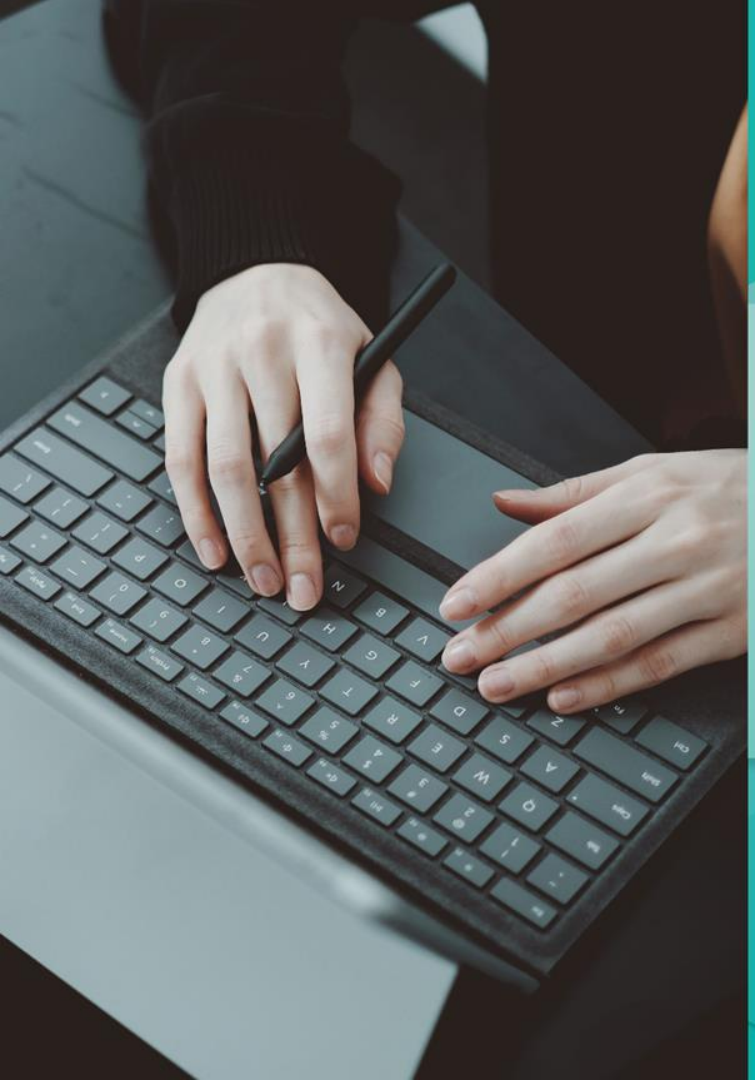

MySkill | #RintisKarirImpian

Case Study

Implementing SQL: TechCorp

FATIMAH SYAHRANI ABDULLAH

Case Study

Implementing SQL: TechCorp

TechCorp adalah perusahaan e-commerce yang berfokus pada penjualan produk elektronik seperti laptop, smartphone, dan aksesoris. TechCorp juga menyediakan layanan dukungan pelanggan untuk membantu pelanggan dengan masalah teknis dan pertanyaan mengenai produk yang dijual.

Sebagai seorang Data Analyst di Perusahaan ini, kamu ditugaskan untuk menganalisis beberapa data untuk keperluan laporan keuangan dan analisis performa bisnis.

Case Study

Implementing SQL: TechCorp

Membuat Tabel

Tabel yang akan dibuat terdiri:

	Tables_in_techcorp
▶	customers
	employees
	orderdetails
	orders
	products
	supportickets
	supporttickets

```
create table products (
    product_id int auto_increment primary key,
    product_name varchar(100) not null,
    category varchar(50),
    price decimal(10,2),
    stock_quantity int
    );
```

```
create table customers (
    customer_id int auto_increment primary k
    first_name varchar(50) not null,
    last_name varchar(50) not null,
    email varchar(50) unique,
    phone varchar(20),
    addres varchar(200)
```

```
create table orderDetails (
    order_detail_id int auto_increment primary key,
    order_id int,
    product_id int,
    quantity int,
    unit_price decimal(10,2),
    foreign key (order_id) references orders(order_id),
    foreign key (product_id) references products(product_id)
);
```

```
create table orders (
    order_id int auto_increment primary key,
    customer_id int,
    order_date date,
    total_amount decimal(10,2),
     foreign key (customer_id) references customers(customer_id)
);
```

```
create table employees (
    employee_id int auto_increment primary key,
    first_name varchar(50),
    last_name varchar(50),
    email varchar (50) unique,
    phone varchar (20),
    hire_date date,
    department varchar (50)
    );
```

```
create table SupportTickets (
    ticket_id int auto_increment primary key,
    customer_id int,
    employee_id int,
    issue text,
    statuss varchar(50),
    created_at datetime,
    resolved_at datetime,
    foreign key (customer_id) references customers(customer_id),
    foreign key (employee_id) references employees(employee_id)
);
```

Case Study

Implementing SQL: TechCorp

Mengimput Nilai dalam Tabel:

```
INSERT INTO products (product_name, category, price, stock_quantity, discount)
VALUES
('Laptop Pro 15', 'Laptop', 1500.00, 100, 0),
('Smartphone X', 'Smartphone', 800.00, 200, 0),
('Wireless Mouse', 'Accessories', 25.00, 500, 0),
('USB-C Charger', 'Accessories', 20.00, 300, 0),
('Gaming Laptop', 'Laptop', 2000.00, 50, 10),
('Budget Smartphone', 'Smartphone', 300.00, 150, 5),
('Noise Cancelling Headphones', 'Accessories', 150.00, 120, 15),
('Wireless Earphones', 'Accessories', 100.00, 100, 10);
```

```
INSERT INTO customers (first_name, last_name, email, phone, addres)
VALUES
('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm Street'),
('Jane', 'Smith', 'jane.smith@example.com', '123-456-7891', '456 Oak Street'),
('Emily', 'Johnson', 'emily.johnson@example.com', '123-456-7892', '789 Pine Street'),
('Michael', 'Brown', 'michael.brown@example.com', '123-456-7893', '101 Maple Street'),
('Sarah', 'Davis', 'sarah.davis@example.com', '123-456-7894', '202 Birch Street');
```

```
INSERT INTO orders (customer_id, order_date, total_amount)
VALUES
(1, '2023-07-01', 1525.00),
(2, '2023-07-02', 820.00),
(3, '2023-07-03', 25.00),
(1, '2023-07-04', 2010.00),
(4, '2023-07-05', 300.00),
(2, '2023-07-06', 315.00),
(5, '2023-07-07', 165.00);
```

```
INSERT INTO orderdetails (order_id, product_id, quantity, unit_price)
VALUES
(1, 1, 1, 1500.00),
(1, 3, 1, 25.00),
(2, 2, 1, 800.00),
(2, 4, 1, 20.00),
(3, 3, 1, 25.00),
(4, 5, 1, 2000.00),
(4, 6, 1, 10.00),
(5, 6, 1, 300.00),
(6, 6, 1, 300.00),
(7, 7, 1, 150.00),
(7, 4, 1, 15.00);
```

```
INSERT INTO employees (first_name, last_name, email, phone, hire_date, department)
VALUES
('Alice', 'Williams', 'alice.williams@example.com', '123-456-7895', '2022-01-15', 'Support'),
('Bob', 'Miller', 'bob.miller@example.com', '123-456-7896', '2022-02-20', 'Sales'),
('Charlie', 'Wilson', 'charlie.wilson@example.com', '123-456-7897', '2022-03-25', 'Development'),
('David', 'Moore', 'david.moore@example.com', '123-456-7898', '2022-04-30', 'Support'),
('Eve', 'Taylor', 'eve.taylor@example.com', '123-456-7899', '2022-05-10', 'Sales');
```

```
INSERT INTO SupportTickets (customer_id, employee_id, issue, statuss, created_at, resolved_at)
VALUES
(1, 1, 'Cannot connect to Wi-Fi', 'resolved', '2023-07-01 10:00:00', '2023-07-01 11:00:00'),
(2, 1, 'Screen flickering', 'resolved', '2023-07-02 12:00:00', '2023-07-02 13:00:00'),
(3, 1, 'Battery drains quickly', 'open', '2023-07-03 14:00:00', NULL),
(4, 2, 'Late delivery', 'resolved', '2023-07-04 15:00:00', '2023-07-04 16:00:00'),
(5, 2, 'Damaged product', 'open', '2023-07-05 17:00:00', NULL),
(1, 3, 'Software issue', 'resolved', '2023-07-06 18:00:00', '2023-07-06 19:00:00'),
(2, 3, 'Bluetooth connectivity issue', 'resolved', '2023-07-07 20:00:00', '2023-07-07 21:00:00'),
(5, 4, 'Account issue', 'open', '2023-07-08 22:00:00', NULL),
(3, 4, 'Payment issue', 'resolved', '2023-07-09 23:00:00', '2023-07-09 23:30:00'),
(4, 5, 'Physical damage', 'open', '2023-07-10 08:00:00', NULL),
(4, 1, 'Laptop blue screen', 'resolved', '2024-01-05 10:00:00', '2024-02-05 12:00:00'),
(5, 1, 'Laptop lagging', 'resolved', '2024-01-06 10:00:00', '2024-01-25 12:00:00'),
(3, 1, 'Some part of laptop broken', 'resolved', '2024-02-05 10:00:00', '2024-03-05 12:00:00');
```

Case Study

Implementing SQL: TechCorp

1. Identifikasi 3 pelanggan teratas berdasarkan total nominal pesanan!

```
select c.first_name, c.last_name, sum(o.total_amount) as total_order
    from customers as c
    join orders as o on o.customer_id = c.customer_id
    group by c.customer_id
    order by total_order desc limit 3;
```

	first_name	last_name	total_order
▶	John	Doe	3535.00
	Jane	Smith	1135.00
	Michael	Brown	300.00

Case Study

Implementing SQL: TechCorp

2. Temukan rata-rata nominal pesanan untuk setiap pelanggan!

```
select c.first_name, c.last_name,
    avg(o.total_amount) as average_order
    from customers as c
    join orders as o on c.customer_id = o.customer_id
    group by c.customer_id;
```

	first_name	last_name	average_order
▶	John	Doe	1767.500000
	Jane	Smith	567.500000
	Emily	Johnson	25.000000
	Michael	Brown	300.000000
	Sarah	Davis	165.000000

Case Study

Implementing SQL: TechCorp

3. Temukan semua karyawan yang telah menyelesaikan lebih dari 4 tiket support!

```
select e.first_name, e.last_name, count(s.ticket_id)
    from employees as e
    join SupportTickets as s on e.employee_id = s.employee_id
    where s.statuss = 'resolved'
    group by e.employee_id
    having count(s.ticket_id) > 4;
```

	first_name	last_name	count(s.ticket_id)
▶	Alice	Williams	5

Case Study

Implementing SQL: TechCorp

4. Temukan semua produk yang belum pernah dipesan!

```
select products.product_name from products
    left join orderdetails on orderdetails.product_id = products.product_id
    where orderdetails.order_id is null;
```

	product_name
▶	Wireless Earphones
	Laptop Pro 15
	Smartphone X
	Wireless Mouse
	USB-C Charger
	Gaming Laptop
	Budget Smartphone
	Noise Cancelling Headphones
	Wireless Earphones

Case Study

Implementing SQL: TechCorp

5. Hitung total pendapatan yang dihasilkan dari penjualan produk!

```
select sum(quantity*unit_price) as total_revenue from orderdetails
```

	total_revenue
▶	5145.00

Case Study

Implementing SQL: TechCorp

6. Temukan harga rata-rata produk untuk setiap kategori dan temukan kategori dengan harga rata-rata lebih dari $500!

```
with cte_avg_price as (
    select category,
    avg(price) rerata from products
    group by category)
    select * from cte_avg_price where  rerata > 500;
```

	category	rerata
▶	Laptop	1750.000000
	Smartphone	550.000000

Case Study

Implementing SQL: TechCorp

7. Temukan pelanggan yang telah membuat setidaknya satu pesanan dengan total jumlah lebih dari $1000!

```
select * from customers
    where customer_id in
    (select customer_id from orders where total_amount > 1000);
```

	customer_id	first_name	last_name	email	phone	addres
▶	1	John	Doe	john.doe@example.com	123-456-7890	123 Elm Street