```
create database Ultratech_cement_manufacturing_company;
use Ultratech_cement_manufacturing_company;
create table Product (Product_id int primary key,
Product_name varchar(255),
stock_quantity varchar(255),
unit_price varchar(255));
create table Suppliers ( supplier_id int auto_increment primary key,
supplier_name varchar(255),
contact_person varchar(255),
contact_number varchar(255));
alter table Suppliers auto_increment =100;
create table Customers (Customer_id int primary key,
Customet_name varchar(255),
contact_Person varchar(255),
contact_number varchar(255));
create table Orders (Order_id int primary key,
customer_id int,
order_data date,
Ship_date date,
foreign key(customer_id) references Customers(customer_id)
on delete cascade
on update cascade);
create table Order_details (order_details_id int primary key,
order_id int,
product_id int,
Quantity int,
foreign key(order_id) references Orders(Order_id),
foreign key(product_id) references product(product_id)
on delete cascade
on update cascade);

insert into Product (Product_id,Product_name,stock_quantity,unit_price) values
(001,"OPC_Cement_33","2100","Rs_390_perbag"),
(002,"OPC_Cement_43","2510","Rs_395_perbag"),
(003,"OPC_Cement_53","1865","RS_385_perbag"),
(004,"PPC_Cement","1460","Rs_375_perbag"),
(005,"PSC_Cement","1680","Rs_395_perbag"),
(006,"ultratech_tilefixo","980","Rs_264_perbag"),
(007,"ultratech_microkreta","655","Rs_480_perbag"),
(008,"ultratech_readiplast","365","Rs_615_perbag"),
(009,"ultratech_fixoblock","895","Rs_260_perbag"),
(010,"ultratech_powergrout","905","Rs_500_perbag"),
(011,"ultratech_weatherpro","450","Rs_2648_per20lit"),
(012,"ultratech_superstcco","885","Rs_450_perbag");
select * from product;
insert into Suppliers (supplier_name,contact_person,contact_number) values
("shri_sai_traders","Shatabdi_chowk_nagpur","650.505.1876"),
("k&k_brothers","Jaripatka_nagpur","650.508.2836"),
("Raghav_trade_link","Gandhibagh_nagpur","515.123.5555"),
("om_traders","shankar_nagar_nagpur","505.100.5100"),
("kk_traders","katol_road_nagpur","510.110.5202"),
("shrilaxminarayan_traders","Manish_nagar_nagpur","650.500.1825"),
("Arriance_consumer","Kharbi_nagpur","525.125.5225"),
("Ali_cement_house","wardhaman_nagar_nagpur","510.110.5100"),
("Ajanta_hardware","Gandhibagh_nagpur","615.525.5150"),
("om_mahalaxmi_hardware","kharbi_nagpur","530.130.5116"),
("Dhomne_treders","Mhalgi_nagar_nagpur","525.100.2020"),
("Poddar_Enterprise","Pratap_nagar_nagpur","525.110.2400");
select * from suppliers;
insert into customers (Customer_id,Customet_name,contact_person,contact_number)
values
(50,"Rajesh","Shankar_nagar_nagpur","9845655521"),
(51,"Shubham","Pratap_nagar_nagpur","9885654522"),
(52,"sunil","laxmi_nagar_nagpur","9695852236"),
(53,"Kartik","Panase_layout_nagpur","9896982255"),
(54,"Aditya","Trimurti_nagar_nagpur","9875652233"),
(55,"sharad","Wardha","9765452245"),
(56,"Hement","Pardi_nagpur","8845652312"),
(57,"Akshay","Manish_nagar_nagpur","8855254565"),
(58,"mrunali","wadi_nagpur","8855654522"),
(59,"kailash","wanadongri_nagpur","8822456525"),
(60,"vijay","kato_road,nagpur","9885759522"),
(61,"Ankita","manewada_nagpur","8855351585"),
(62,"sagar","Khamla_nagpur","8875654522");
select * from customers;



insert into Orders (order_id,customer_id,order_data,ship_date) values
(10,50,"2023-04-10","2023-04-12"),
(11,51,"2023-04-15","2023-04-17"),
(12,52,"2023-05-02","2023-05-04"),
(13,53,"2023-05-25","2023-05-26"),
(14,54,"2023-06-12","2023-06-13"),
(15,55,"2023-07-10","2023-07-13"),
(16,56,"2023-07-22","2023-07-24"),
(17,57,"2023-07-27","2023-07-29"),
(18,58,"2023-08-05","2023-08-06"),
(19,59,"2023-08-18","2023-08-21"),
(20,60,"2023-09-10","2023-09-15"),
(21,61,"2023-09-22","2023-09-24"),
(22,62,"2023-10-12","2023-10-13");
select * from orders;
insert into order_details (order_details_id,order_id,product_id,Quantity) value
(1201,10,001,962),
(1202,11,002,2100),
(1203,12,003,1445),
(1204,13,004,995),
(1205,14,005,1220),
(1206,15,006,450),
(1207,16,007,350),
(1208,17,008,110),
(1209,18,009,410),
(1210,19,010,495),
(1211,20,011,125),
(1212,21,012,435);

select * from product;
select * from suppliers;
select * from customers;
select * from orders;
select * from order_details;

# 1 view product info with order details ?
create view detail as
select product.Product_id, product.Product_name, product.stock_quantity,
product.unit_price,
order_details.order_details_id, order_details.order_id, order_details.Quantity from
product
join order_details on product.product_id = order_details.product_id;
select * from detail;

# 2 Manage supplier and customer using add, update, delete
  # add column city in customers table
alter table customers
add column city varchar(255);
select * from customer;
  # update customer customer contact number in customers table
update customers
set contact_number = 8830895725
where customer_id = 54;
select * from customers;
  # Delete outdated suppliers who cannot purchase UltraTech Cement products
SET SQL_SAFE_UPDATES = 0;
delete from suppliers
where supplier_name = "Poddar_Enterprise";

# 3 Generate report -product sales order history and current inventory
#Ans:- product sales order history is Total Quantity of product sales and
#      current inventory is the total product stock quantity currently available in
it
select sum(product.stock_quantity) as Total_availabe_stock,
sum(order_details.Quantity) as Total_sales_stock from product
join order_details on product.product_id = order_details.product_id;

# 4 Supplier management include with contact details
select contact_person, contact_number from suppliers;

# 5 write a sql queery to retreieve stock order with the associate customer name
```

```
select sum(order_details.quantity) as total_stock_order, customers.Customet_name
from customers
join Orders on customers.customer_id = Orders.customer_id
join order_details on Orders.order_id = order_details.order_id
Group by customers.Customet_name;

# 6 create a query to display the product ordered in a specific oreder including
their name and quantities

select product.product_id, product.product_name, order_details.Quantity
from product
join order_details on product.product_id = order_details.product_id
order by product.product_id asc;

# 7 Provide SQL statements to add list  of ordered in a specific order including
their name and quantities

select order_details.order_id, product.product_name,  order_details.Quantity
from product
join order_details on product.product_id = order_details.product_id
order by order_details.order_id asc;

# 8 Shows the relationship between order and order_details
# Ans:- order table order_id primary key linked with order_details foreign key with
same data type and data
#       this relation shows also in Entity-relationship (EER) diagram
#       Step to show EER diagram (First click database in worbanch after select
reverse engg and then click 2 times next and select database
#       and execute after that show entity diagram)

# 9 How do you handle or exception during the insertion of new record into a order
and order details
# Ans:-Handling exceptions during the insertion of new records into the Order and
OrderDetails tables in a database involves implementing proper error
#      handling mechanisms to ensure data integrity and reliability use
transaction, try catch_block, rolback on error, error logging or notification

# 10 view solution how would you secure the database from unauthorised access
# Ans:- Securing a database from unauthorized access is a critical aspect of
maintaining data integrity, confidentiality, and availability.
#       Implementing a comprehensive security strategy involves multiple layers and
practices use Authentication that is  Strong Password Policies
#       Enforce strong password policies requiring users to create complex
passwords that include a mix of alphanumeric characters, symbols, and a minimum
length.

# 11 Describe the steps to restore the database to a specific time in time using
backups
# Ans:- Restore a database to a specific point in time using backups
#       Restore the full backup, followed by transaction log backups up to the
```

```
desired time using the appropriate commands for your database system.

# 12 Explain the challenges of handling concurens transaction in the database where
multiple users may be placing orders simultenius.
# Ans:- Handling concurrent transactions in a database where multiple users are
placing orders simultaneously introduces challenges like:
#        Data Conflicts: Users updating the same data simultaneously can lead to
conflicts and inconsistencies.
#       Lost Updates: Changes made by one user may be overwritten by another,
causing data loss.
#       Deadlocks: Transactions may get stuck in a deadlock, where each is waiting
for the other to release a lock.
#       Performance Impact: Implementing locks for data consistency may slow down
the system due to increased contention.
#       Isolation Levels: Balancing between high isolation for data consistency and
low isolation for better performance is challenging.
#        Rollback Complexity: Handling rollbacks when conflicts occur adds
complexity to the transaction management process.
#       Scalability Issues: In distributed systems, managing concurrent
transactions across multiple locations introduces additional coordination
challenges.

# 13 Design a query to calculate the total quantity of specific product sold across
all orders

select sum(order_details.Quantity) as total_quantity_product_sold,
product.product_name
from product
join order_details on product.product_id = order_details.product_id
Group by product.product_name;

# 14 Design a query to generate a report showing the sales of each project

select sum(order_details.Quantity) as total_quantity_product_sold,
product.product_name, product.product_id
from product
join order_details on product.product_id = order_details.product_id
Group by product.product_name, product.product_id;

# 15 combine the orders and order details table using inner join and filter the
result display details for a specific order

select orders.order_id, orders.customer_id, orders.order_id, orders.ship_date,
order_details.order_details_id, order_details.product_id,
order_details.Quantity from orders
inner join order_details on orders.order_id = order_details.order_id
order by orders.order_id;

# 16 Design a query group by customer and calculate the total quantity of products
order by each customers
```

```
select sum(Order_details.Quantity) as total_quantity_products,
customers.customet_name from product
join Order_details on product.product_id = Order_details.product_id
join orders on Order_details.order_id = orders.order_id
join customers on orders.customer_id = customers.customer_id
Group by customers.customet_name;

# 17 Utilize subquery to retrive the name who have place order

SELECT customet_name FROM customers
WHERE customer_id IN (SELECT customer_id FROM orders);

# 18 combine the product order and order details to retrive the list of products
order by specific customer

select Order_details.Quantity, customers.customet_name from product
join Order_details on product.product_id = Order_details.product_id
join orders on Order_details.order_id = orders.order_id
join customers on orders.customer_id = customers.customer_id;

# 19 implement a query to retrive the first 10 product from product table

select * from product limit 10;

# 20 ultilise the subquery to find product with a stock quantity greater than the
average stock quantity of all product

select stock_quantity from product
where stock_quantity > (select avg(stock_quantity) from product);
```