Queries optimization

simple query	Execution time before optimization	Optimization technique	Rewrite query	Execution time after optimization
`select category_id , count(product_id) as products_num from product group by categroy_id`	1937.361 ms	covering index on category id and materialized view	`createindex idx_covering_category_product ON product (product_id) include (category_id) CREATE MATERIALIZED VIEW categroy_products AS select category_id , count(product_id) as products_num from product group by category_id;`	8.033 ms
`SELECT customer_id, SUM(total_amount) AS total_spending FROM orders GROUP BY customer_id ORDER BY total_spending DESC LIMIT 10;`	14105.492 ms	covering index and materialized views	`CREATE INDEX idx_covering_customer_total_product ON orders (customer_id) include (total_amount); CREATE MATERIALIZED VIEW customer_total_spending AS SELECT customer_id, SUM(total_amount) AS total_spending FROM orders GROUP BY customer_id ORDER BY total_spending DESC LIMIT 10; EXPLAIN ANALYZE SELECT * FROM customer_total_spending;`	0.059 ms
`select product_name , stock_quantity from product where stock_quantity < 3;`	386.295 ms	B-tree index and materialized view	`create index idx_stock_quantity on product(stock_quantity) create materialized view low_stock_quantity as select product_name , stock_quantity from product`	5.260 ms

simple query	123 Execution time before optimization	Optimization technique	Rewrite query	Execution time after optimization
			where stock_quantity < 3; explain analyze select * from low_stock_quantity;	
SELECT c.category_name, SUM(oi.quantity * oi.unit_price) AS total_revenue FROM category c JOIN product p ON c.category_id = p.category_id JOIN order_details oi ON p.product_id = oi.product_id GROUP BY c.category_id, c.category_name ORDER BY total_revenue DESC;	21483.389 ms	composite index and materialized view	create index idx_cat_prod on product(product_id , category_id) create materialized view cat_total_revenue as SELECT c.category_name, SUM(oi.quantity * oi.unit_price) AS total_revenue FROM category c JOIN product p ON c.category_id = p.category_id JOIN order_details oi ON p.product_id = oi.product_id GROUP BY c.category_id, c.category_name ORDER BY total_revenue DESC; explain analyze select * from cat_total_revenue;	0.0253 ms