OLAP Queries:

-> Find number of products in every category

```
select category.cat_name, product.prod_name, sum(product.stock) as sum
from category,product
where product.cat_id = Category.cat_id
group by category.cat_name, product.prod_name with rollup;
```

-> Find number of products in every brand

```
select brand.brand_name, product.prod_name, sum(product.stock) as sum
from brand,product
where product.brand_id = brand.brand_id
group by brand.brand_name, product.prod_name with rollup;
```

-> Find number orders made by every customer

```
select orders.cust_id, count(orders.order_id)
from orders
group by orders.cust_id with rollup;
```

-> Find total sum of every order

```
select orders.order_id,orders.cust_id, sum(orders.total)
from orders
group by orders.cust_id,orders.order_id with rollup;
```

Triggers:

```
delimiter //
CREATE TRIGGER update_placed_products_cost BEFORE INSERT ON placed_products
FOR EACH ROW
BEGIN


   SET NEW.total_cost = (
         SELECT o.total - (c.discount/100)*o.total as after_discount
         FROM Orders o
         JOIN Coupon c ON o.coup_id = c.coup_id
         WHERE o.order_id = New.order_id
         );


END;//
```

This trigger updates the value of total_cost right after discount on placed_products table.

```
delimiter //
CREATE TRIGGER new_customer_coupon
AFTER INSERT ON Customers
FOR EACH ROW
BEGIN
  INSERT INTO has_coupon (cust_id, coup_id)
  VALUES (NEW.cust_id, 1);
END;//
```

This trigger gives a newly signed customer a free coupon with 5% discount.

```
delimiter //
create trigger age
after insert on customers
for each row
begin
         if new.phone_numb < 1000000000 then
                  signal sqlstate '45000' set message_text = 'Invalid phone number';
         end if;
end;//
```

This trigger throws an error if a customer registers invalid mobile number.

Embedded Queries:

Queries are embedded in JavaScript:

1. This query lists returns all brands.

```
const getAllBrand = (req,res) =>{

    con.query('SELECT * FROM brand;',(err,results)=>{
        if(err) throw err;
        res.status(201).send(results);
    })


}
```

2. This query lists returns all categories.

```
const getAllCategory = (req,res) =>{

    con.query('SELECT * FROM category;',(err,results)=>{
        if(err) throw err;
        res.status(201).send(results);
    })


}
```

3. 1. This query lists returns all products.

```
const getAllProducts = (req,res) =>{

    con.query('SELECT * FROM product;',(err,results)=>{
        if(err) throw err;
        res.status(201).send(results);
    })



}
```