Trigger 1:

Assigns 3 stars to a new delivery partner, whenever he/she is added to the database.

CREATE TRIGGER delivery\_partner\_rating

BEFORE INSERT ON delivery\_partner

FOR EACH ROW

SET NEW.partner\_rating = 3;

Check if the query is working:

INSERT INTO delivery\_partner (idDelivery\_partner,partner\_name, partner\_contact,order\_id) VALUES (21,'Test Partner', 1234567890,21);

SELECT \* FROM delivery\_partner where idDelivery\_partner=21;

Trigger 2:

Changes the total price of the cart whenever a new product is added.

Check if the query is working:

INSERT INTO products\_in\_cart (cart\_id\_inside, product\_id\_inCart) VALUES (2, 2);

SELECT \* FROM badibasket.cart;

DELIMITER //

CREATE TRIGGER update\_cart\_total

AFTER INSERT ON products\_in\_cart

FOR EACH ROW

BEGIN

UPDATE cart SET total\_price = (

SELECT SUM(product.product\_price)

FROM products\_in\_cart

INNER JOIN product ON products\_in\_cart.product\_id\_inCart = product.idProduct

WHERE products\_in\_cart.cart\_id\_inside = NEW.cart\_id\_inside

) WHERE idCart = NEW.cart\_id\_inside;

END //

DELIMITER ;

OLAP Queries:  
  
  
1. This query returns the total sales of during a year (it also prints quarterly sales).

SELECT YEAR(estimated\_time\_of\_delivery) AS year, QUARTER(estimated\_time\_of\_delivery) AS quarter, SUM(total\_amount) AS sales

FROM `order`

GROUP BY YEAR(estimated\_time\_of\_delivery), QUARTER(estimated\_time\_of\_delivery) WITH ROLLUP;  
  
  
2. This query returns the year-wise expenditure of a customer.  
  
SELECT customer.customer\_name, YEAR(estimated\_time\_of\_delivery) AS year, SUM(total\_amount) AS sales

FROM `order`

INNER JOIN customer ON `order`.customer\_id = customer.idCustomer

GROUP BY customer.customer\_name, YEAR(estimated\_time\_of\_delivery) WITH ROLLUP;

3. This query returns the total number of orders placed by a customer:  
  
SELECT

c.customer\_name,

COUNT(o.idOrder) AS num\_orders

FROM

customer c

LEFT JOIN `order` o ON c.idCustomer = o.customer\_id

GROUP BY

c.customer\_name

WITH ROLLUP;

4. This query returns the average order amount for each address.

SELECT

address,

YEAR(estimated\_time\_of\_delivery) AS order\_year,

AVG(total\_amount) AS avg\_order\_amount

FROM

`order`

INNER JOIN customer ON customer\_id = idCustomer

GROUP BY

address,

order\_year WITH ROLLUP;