Jennifer Hann
Dipesh Patel

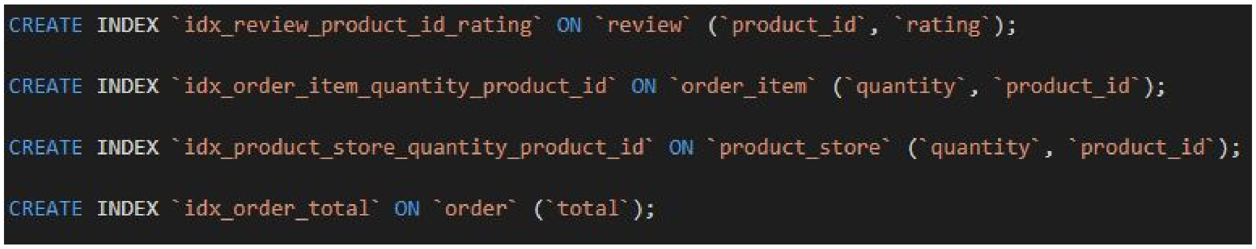
Database II: Assignment 3

Changes to be made

On the general, our database was well which can be used even with our issues.
There weren't many issues, rather only 1 major and 2 minor. Our indexes weren't properly made; we didn't properly implement the concept of indexes. One of our roles was missing a bit of permission and one of our triggers seemed somewhat useless which could be avoided by adding an attribute to a table.

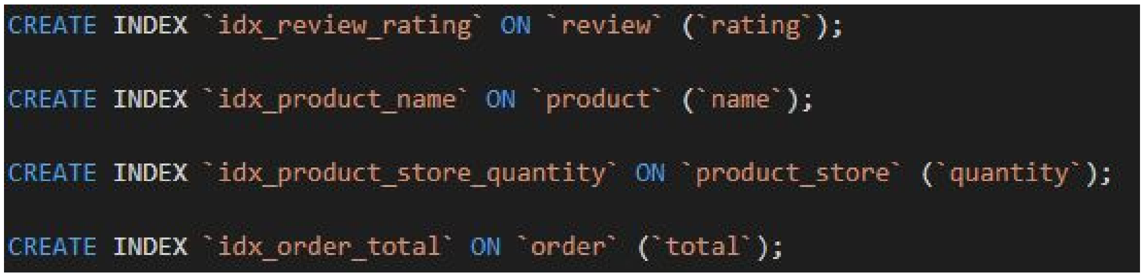
Index

Initially our indexes were the following:



```
CREATE INDEX `idx_review_product_id_rating` ON `review` (`product_id`, `rating`);

CREATE INDEX `idx_order_item_quantity_product_id` ON `order_item` (`quantity`, `product_id`);

CREATE INDEX `idx_product_store_quantity_product_id` ON `product_store` (`quantity`, `product_id`);

CREATE INDEX `idx_order_total` ON `order` (`total`);
```

An index is supposed to sort on a field that is different than the primary key because that would just be redentund and efficient. So we changed a few things. We got rid of all ids in the indexes. We also rethinked about which index we need since it would be smart to put indexes about queries that are frequent. So we added an index on the product name in the product table. Ans now this is our new indexes:

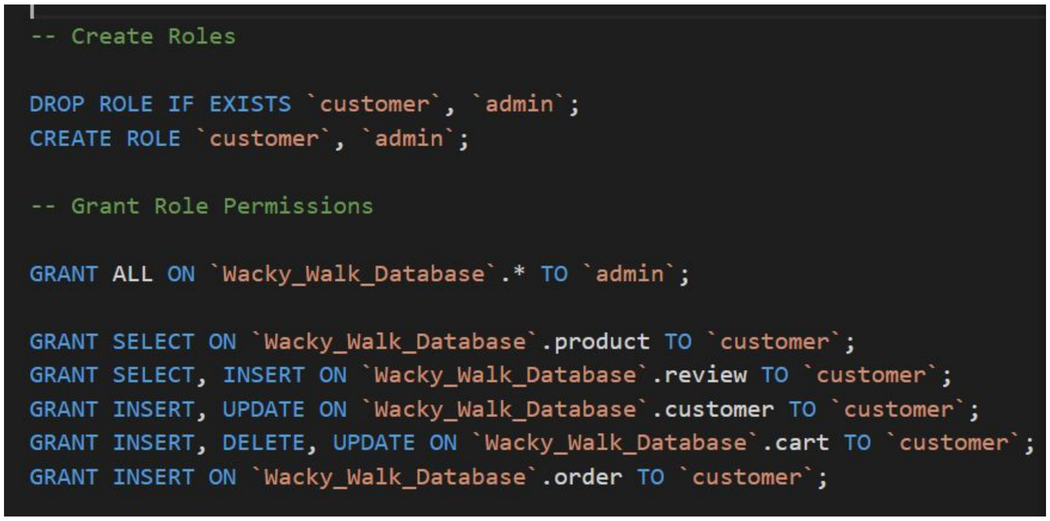


```
CREATE INDEX `idx_review_rating` ON `review` (`rating`);

CREATE INDEX `idx_product_name` ON `product` (`name`);

CREATE INDEX `idx_product_store_quantity` ON `product_store` (`quantity`);

CREATE INDEX `idx_order_total` ON `order` (`total`);
```

Permissions

For the second assignment, we only had one role that could use the whole database. We didn't make roles properly that would limit the user from accessing tables that were not meant to them like the customer that is using the website to only shop. We created two roles for the user of our webpage, one of them is the admin who has access to all the tables in the database. The person that has the admin role is the store owner and the second role is for the customer.

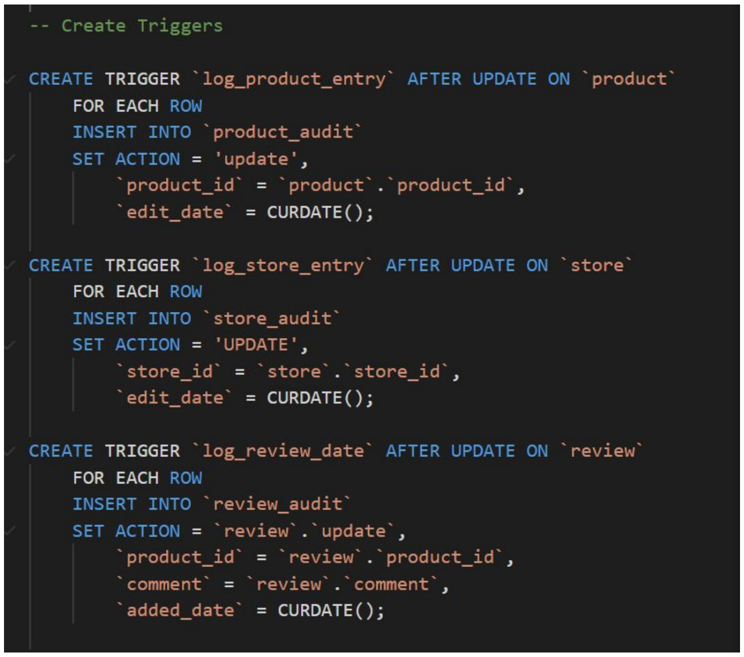
Customers can only access tables that allow them to see the products, the reviews, their account information, their cart and their order.



```sql
-- Create Roles

DROP ROLE IF EXISTS `customer`, `admin`;
CREATE ROLE `customer`, `admin`;

-- Grant Role Permissions

GRANT ALL ON `Wacky_Walk_Database`.* TO `admin`;

GRANT SELECT ON `Wacky_Walk_Database`.product TO `customer`;
GRANT SELECT, INSERT ON `Wacky_Walk_Database`.review TO `customer`;
GRANT INSERT, UPDATE ON `Wacky_Walk_Database`.customer TO `customer`;
GRANT INSERT, DELETE, UPDATE ON `Wacky_Walk_Database`.cart TO `customer`;
GRANT INSERT ON `Wacky_Walk_Database`.order TO `customer`;
```

Triggers

Our first and second trigger run on an UPDATE trigger. It could be smart to put it on an INSERT because the triggers are logging the date of when the product and store. But we put it on an UPDATE trigger because we would also keep track of when the product and store were removed from the list. This could help track the period of time between the owners of the company and the store. This could also help remember when a product was removed and if it should be readded. As for the third trigger, we could remove it from adding a date field to the review table but that would overload the table since that table also holds pictures and videos.To make it easier for the admins to find the date of each comment we put it in a different table.



```sql
-- Create Triggers

CREATE TRIGGER `log_product_entry` AFTER UPDATE ON `product`
    FOR EACH ROW
    INSERT INTO `product_audit`
    SET ACTION = 'update',
        `product_id` = `product`.`product_id`,
        `edit_date` = CURDATE();

CREATE TRIGGER `log_store_entry` AFTER UPDATE ON `store`
    FOR EACH ROW
    INSERT INTO `store_audit`
    SET ACTION = 'UPDATE',
        `store_id` = `store`.`store_id`,
        `edit_date` = CURDATE();

CREATE TRIGGER `log_review_date` AFTER UPDATE ON `review`
    FOR EACH ROW
    INSERT INTO `review_audit`
    SET ACTION = `review`.`update`,
        `product_id` = `review`.`product_id`,
        `comment` = `review`.`comment`,
        `added_date` = CURDATE();
```