EBD: Database Specification Component

Our website aims to help customers get what then need when they need it during these troubling times with an enjoyable browsing experience.

A4: Conceptual Data Model

The Fneuc shop website provides a reliable shopping service for the general public with easy access.
This artefact contains the conceptual data model for the platform, including business rules.

1. Class diagram

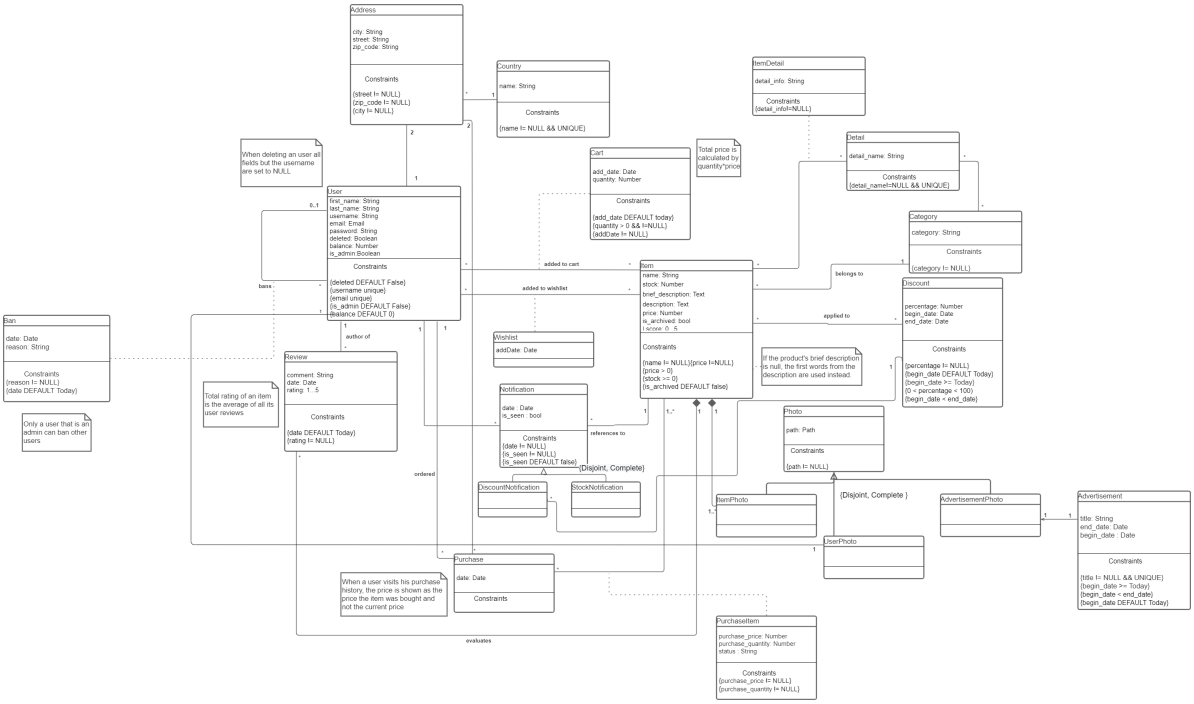

Figure 1: Class Diagram.

2. Additional Business Rules

Additional business rules are represented as UML notes in the class diagram.

A5: Relational Schema, validation and schema refinement

In this artifact our website's database relational schema is presented as well as it's valitation and refinements.

1. Relational Schema

Relation referenct	Relation Compact Notation
R01	country(country_id, name **NN**)
R02	address(address_id, city **NN**, street **NN**, zip_code **NN**, country→country)
R03	photo(photoID, path **NN**)
R04	users(user_id, first_name , last_name, username **UK** , email **UK**, password, billing_addr→address, shipping_addr→address,photo_id→photo, deleted **DF** False, balance **DF** 0, is_admin **DF** False)
R05	review(review_id, user_id→users, comment, date **DF** Today, rating **NN CK** rating > 0 AND rating < = 5)
R06	category(category_id, name **UK**)
R07	detail(detail_id, name **NN UK**)
R08	item(item_id, name **NN**, stock **NN CK** stock >= 0, brief_description, description **NN**, price **NN CK** price> 0, is_archived **DF** False, category→category, score)
R09	ban(admin_id→users, user_id→users, date **DF** Today, reason **NN**)
R10	purchase(purchase_id, user_id→users, date **DF** Today)
R11	purchase_item(purchase_id→purchase, item_id→item, price **NN**, quantity **NN**)
R12	advertisement(advertisement_id, title **NN UK**, begin_date **DF** Today, end_date **CK** end_date > begin_date, photo_id→photo)
R13	item_photo(photo_id→photo, item_id→item)
R14	cart(user_id→user, item_id→item, add_date **DF** Today, quantity **CK** quantity > 0)
R15	wishlist(user_id→user, item_id→item, add_date **DF** Today)
R16	discount(discount_id, percentage **CK** percentage > 0 && percentage < 100, begin_date **DF** Today, end_date **CK** end_date > begin_date)
R17	notification(notification_id, user_id→user, item_id→item, discount_id→discount,date **DF** Today, is_seen **DF** False, type)
R18	apply_discount(item_id→item, discount_id→discount)
R19	item_detail(item_id→item, detail_id→detail, detail_info **NN**)
R20	category_detail(category_id→category, detail_id→detail)

All users' attributes must be not null when deleted is false but all but the id are null when deleted is true. Because of this, the attributes cannot have the flag not null.
Generalizations:

- Authenticated/Admin: Added a boolean to represent an admin as creating a new table with no fields is not recommended in this case.
- Types of notifications: Merged them together as they have the same atributes except for one.

- Photos: Since the relation for the photos of users and advertisements is 1-1, it better if the user and the advertisement keep a reference to their photo.

2. Domains

Domain Name	Domain Specification
Today	DATE DEFAULT CURRENT_DATE
TYPE	ENUM ('Discount', 'Stock')

3. Schema validation

In this section, the functional dependencies and the normal form each table is in is presented

TABLE R01	**users**
Keys	{ user_id }
Functional Dependencies:	
FD0101	{ user_id } → { email, name, username, password, billing_addr, shipping_addr,photo_id,deleted, balance, is_admin}
NORMAL FORM	BCNF

TABLE R02	**review**
Keys	{ review_id }
Functional Dependencies:	
FD0401	{ review_id } → { user_id, comment, date, rating }
NORMAL FORM	BCNF

TABLE R03	**category**
Keys	{ category_id }
Functional Dependencies:	
FD0501	{ category_id} → { name }
NORMAL FORM	BCNF

TABLE R04	detail
Keys	{ detail_id }
Functional Dependencies:	
FD0601	{ detail_id } → { name }
NORMAL FORM	BCNF

TABLE R05	item
Keys	{ item_id }
Functional Dependencies:	
FD0701	{ item_id } → { name, stock, description, price, brief_description, price, is_archived, category}
NORMAL FORM	BCNF

TABLE R06	ban
Keys	{ admin_id, user_id }
Functional Dependencies:	
FD0801	{ admin_id, user_id } → { date, reason }
NORMAL FORM	BCNF

TABLE R07	country
Keys	{ country_id }
Functional Dependencies:	
FD0901	{ country_id } → { name }
NORMAL FORM	BCNF

TABLE R8	address
Keys	{ address_id }
Functional Dependencies:	
FD1001	{ address_id } → { city, street, zip_code, country }
NORMAL FORM	BCNF

TABLE R9	purchase
Keys	{ purchase_id }
Functional Dependencies:	
FD1101	{ purchase_id } → { date }
NORMAL FORM	BCNF

TABLE R10	purchase_item
Keys	{ purchase_id, item_id }
Functional Dependencies:	
FD1201	{ purchase_id, item_id } → { purchase_price, quantity }
NORMAL FORM	BCNF

TABLE R11	photo
Keys	{ photo_id }
Functional Dependencies:	
FD1101	{ photo_id } → { path }
NORMAL FORM	BCNF

TABLE R12	advertisement
Keys	{ advertisement_id }, { title }
Functional Dependencies:	
FD1201	{ advertisement_id} → { title, begin_date, end_date }
FD1202	{ title } -> { advertisement_id, begin_date, end_date }
NORMAL FORM	BCNF

TABLE R13	item_photo
Keys	{ photo_id }
Functional Dependencies:	
FD1301	{ photo_id } → { item_id }
NORMAL FORM	BCNF

TABLE R14	cart
Keys	{ user_id, item_id }
Functional Dependencies:	
FD1401	{ user_id, item_id } → { add_date, is_seen, type }
NORMAL FORM	BCNF

TABLE R15	wishlist
Keys	{ user_id, item_id }
Functional Dependencies:	
FD1501	{ user_id, item_id } → { add_date }
NORMAL FORM	BCNF

TABLE R016	notification
Keys	{ user_id, item_id }
Functional Dependencies:	
FD1601	{ user_id, item_id } → {item_id, discount_id,date, is_seen, type }
NORMAL FORM	BCNF

TABLE R17	discount
Keys	{ discount_id }
Functional Dependencies:	
FD1701	{ discount_id } → { percentage, begin_date, end_date }
NORMAL FORM	BCNF

TABLE R18	apply_discount
Keys	{ item_id, discount_id }
Functional Dependencies:	
(none)	
NORMAL FORM	BCNF

TABLE R19	item_detail
Keys	{ item_id, detail_id }
Functional Dependencies:	
FD1901	{ item_id, detail_id } -> { detail_info }
NORMAL FORM	BCNF

TABLE R20	category_detail
Keys	{ category_id, detail_id }
Functional Dependencies:	
FD2001	{ item_id, detail_id } -> { detail_info }
NORMAL FORM	BCNF

A6: Indexes, triggers, user functions, transactions and population

This artifact cointais all indexes, triggers, functions and transactions needed for the database to work as well as the estimated growth of the database.

1. Database Workload

1.1. Tuple Estimation

Relation reference	Relation Name	Order of magnitude	Estimated growth
R01	country	hundreds	none
R02	photo	thousands	units per day
R03	address	thousands	units per day
R04	users	thousands	dozens per day
R05	review	tens of thousands	hundreds per day
R06	details	hundreds	units per year
R07	category	units	units per year
R08	item	thousands	units per month
R09	ban	hundreds	units per week
R10	purchase	thousands	units per day
R11	purchaseItem	tens of thousands	dozens per day
R12	advertisement	thousands	units per week
R13	itemPhoto	thousands	units per day
R14	cart	tens of thousands	dozens per day
R15	wishlist	tens of thousands	dozens per day
R16	discount	hundred	units per month
R17	notification	tens of thousands	dozens per day
R18	discountNotification	thousands	dozens per day
R19	itemDetail	tens of thousands	dozens per month
R20	categoryDetail	hundreds	dozens per year

1.2. Frequent Queries

Query reference	SELECT01
Query description	get user info
Query frequency	thousands per day

```
SELECT * FROM users WHERE user_id = $id;
```

Query reference	SELECT02
Query description	get user shipping address
Query frequency	hundreds per day

```
SELECT addr.*
FROM users JOIN
(
    FROM address JOIN country USING (country_id)
) AS addr ON (users.shipping_address = addr.address_id)
WHERE users.user_id = $id
```

Query reference	SELECT03
Query description	get user billing address
Query frequency	hundreds per day

```
SELECT addr.*
FROM users JOIN
(
    FROM address JOIN country USING (country_id)
) AS addr ON (users.billing_address = addr.address_id)
WHERE users.user_id = $id
```

Query reference	SELECT04
Query description	sign in
Query frequency	thousands per day

```
#login using username
SELECT user_id FROM USER
WHERE username = $username AND password = $pwd

#login using email
SELECT user_id FROM USER
WHERE email = $username AND password = $pwd
```

Query reference	SELECT05
Query description	get item's info
Query frequency	tens of thousands per day

```
SELECT item.item_id, details_info.detail_info, details_info.name
FROM item
JOIN
(item_detail JOIN details USING (detail_id ))  AS details_info
USING (item_id)
WHERE item.item_id = $item_id
```

Query reference	SELECT06
Query description	search for an item that is not archived
Query frequency	tens of thousands per day

```
SELECT *, ts_rank_cd(search, query) FROM item, to_tsquery('english', $search) AS
query
WHERE search @@ query AND is_archived = false
```

Query reference	SELECT07
Query description	search for an item in a price range within a cetegory
Query frequency	tens of thousands per day

```
SELECT item.* FROM item
WHERE price < $max_price AND price > $min_price AND category_id = $cat_id AND
item.is_archived = false
```

Query reference	SELECT08
Query description	get a user's wishlist
Query frequency	thousands per day

```
SELECT wishlist_items.*
FROM users JOIN
(wishlist JOIN item USING (item_id)) AS wishlist_items
ON (wishlist_items.user_id = users.user_id)
WHERE users.user_id = $usr_id
```

Query reference	SELECT09
Query description	get a user's cart
Query frequency	thousands per day

```
SELECT cart_items.*
FROM users JOIN
(cart INNER JOIN item USING (item_id)) AS cart_items
ON (cart_items.user_id = users.user_id)
WHERE users.user_id = $usr_id
```

Query reference	SELECT10
Query description	all detail's name of a category
Query frequency	hundreds per day

```
SELECT detail_name.name
FROM category JOIN (category_detail JOIN details USING (detail_id)) AS
detail_name USING (category_id)
WHERE category.category_id = $cat_id
```

Query reference	SELECT11
Query description	categories for dropdown filter
Query frequency	thousands per day

```
SELECT category.name
FROM category
WHERE category.category_id = $cat_id
```

Query reference	SELECT12
Query description	get items reviews
Query frequency	thousands per day

```
SELECT users.username, item_reviews.comment_text, item_reviews.date,
item_reviews.rating
FROM users JOIN (review JOIN item USING (item_id)) AS item_reviews USING
(user_id)
WHERE item_reviews.item_id = $item_id
```

Query reference	SELECT13
Query description	get user's notifications
Query frequency	thousands per day

```
SELECT notification_item.*
FROM users JOIN (
    notification JOIN item USING (item_id)) AS notification_item USING (user_id)
WHERE users.user_id = $usr_id
```

Query reference	SELECT14
Query description	search for item's discount
Query frequency	thousands per day

```
SELECT appliable_discount.*
FROM item JOIN (apply_discount JOIN discount USING (discount_id)) AS
appliadble_discount USING (item_id)
WHERE item.item_id = $itm_id
```

Query reference	SELECT15
Query description	get all available discounts
Query frequency	thousands per day

```
SELECT *
FROM advertisement
WHERE begin_date >= now()::date AND end_date <= now()::date
```

Query reference	SELECT16
Query description	get all discounts
Query frequency	thousands per day

```
SELECT *
FROM advertisement
```

Query reference	SELECT17
Query description	user's purchase history
Query frequency	thousands per day

```
SELECT prcs_items.*
FROM users JOIN (purchase JOIN purchase_item USING (purchase_id)) AS prcs_items
USING(user_id)
WHERE users.user_id = $usr_id
```

Query reference	SELECT18
Query description	get all of an item's photos
Query frequency	thousands per day

```
SELECT item_photos.path
FROM item JOIN (item_photo JOIN photo USING (photo_id)) as item_photos USING
(item_id)
WHERE item.item_id = $itm_id
```

Query reference	SELECT19
Query description	get the user's top 3 most frequent categories purchases
Query frequency	thousands per day

```
select category.name, count(*) as occurrences
from category join
(
    --gets users bought items
    item join
    (
        -- gets user's bought items ids
        purchase_item join
        (users join purchase using (user_id)) as user_purchases using
(purchase_id)
    ) as bought_items_ids using(item_id)
```

```
) as bought_items using (category_id)
where bought_items.user_id = $user_id
group by (category.name)
order by occurrences desc
limit 3;
```

1.3. Frequent Updates

Query	UPDATE01
Description	Update user information
Frequency	hundred per month

```
UPDATE "user"
    SET first_name = $first_name, last_name = $last_name, email = $email,
    password = $password, billingAddress = $billingAddress,
    shippingAddress = $shippingAddress, photoID = $photoID
    WHERE userID = $userID
```

Query	UPDATE02
Description	Update item information
Frequency	hundred per month

```
UPDATE "item"
    SET stock = $stock, brief_description = $brief_description, description =
$description,
    price = $price, isArchived = $isArchived, category = $category
    WHERE itemID = $itemID
```

Query	UPDATE03
Description	Update cart information
Frequency	hundred per month

```
UPDATE "cart"
    SET addDate = $addDate, quantity = $quantity
    WHERE userID = $userID AND itemID = $itemID
```

Query	INSERT01
Description	New user registered
Frequency	dozens per day

```
INSERT INTO "user" (first_name,last_name,username,email,password)
    VALUES($first_name,$last_name,$username,$email,$password)
```

Query	INSERT02
Description	New item for sale
Frequency	hundreds per month

```
INSERT INTO "item" (name,stock,brief_description,description,price,category)
    VALUES ($name,$stock,$brief_description,$description,$price,$category)
```

Query	INSERT03
Description	Create new review
Frequency	hundreds per month

```
INSERT INTO "review" (userID,comment,date,rating)
    VALUES ($userID,$comment,$date,$rating)
```

Query	INSERT04
Description	Create new address
Frequency	hundreds per month

```
INSERT INTO "address" (city,street,zip_code,country)
    VALUES ($city,$street,$zip_code,$country)
```

Query	INSERT05
Description	Ban user
Frequency	dozens per month

```
INSERT INTO "ban" (adminID,userID,date,reason)
    VALUES ($adminID,$userID,$date,$reason)
```

Query	INSERT06
Description	Make new purchase
Frequency	dozens per day

```
INSERT INTO "purchase" (userID,date)
    VALUES ($userID,$date)
```

Query	INSERT07
Description	Add item to cart
Frequency	dozens per day

```
INSERT INTO "cart" (user_id,item_id,add_date,quantity)
    VALUES ($user_id,$item_id,$add_date,$quantity )
```

Query	INSERT08
Description	Add item to wishlist
Frequency	dozens per day

```
INSERT INTO "wishlist" (user_id,item_id,add_date)
    VALUES ($user_id,$item_id,$add_date)
```

Query	INSERT09
Description	New notification
Frequency	dozens per day

```
INSERT INTO "notification" (user_id,item_id,discount_id,date,is_seen,type)
    VALUES ($user_id,$item_id,$discount_id,$date,$is_seen,$type)
```

Query	INSERT10
Description	Discount on category
Frequency	units per week

```
INSERT INTO apply_discount(item_id,discount_id)
    SELECT item_id, $discount_id
    FROM item JOIN category USING (category_id)
    WHERE category.name = $category
```

Query	DELETE01
Description	Remove item from cart
Frequency	dozens per day

```
DELETE FROM "cart"
    WHERE user_id=$user_id AND item_id=$item_id
```

Query	DELETE02
Description	Remove item from wishlist
Frequency	dozens per day

```
DELETE FROM "wishlist"
    WHERE user_id=$user_id AND item_id=$item_id
```

2. Proposed Indices

2.1. Performance Indices

> Indices proposed to improve performance of the identified queries.

Index	IDX01
Related queries	SELECT05
Relation	item_detail
Attribute	item_id
Type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	To allow getting all details of an item. Its clustered to allow getting all the details of an item faster. Cardinality is medium

```
CREATE INDEX item_detail_itemID ON item_detail USING btree(item_id);
CLUSTER item_detail using item_detail_itemID;
```

Index	IDX02
Related queries	SELECT07
Relation	item
Attribute	price
Type	B-tree
Cardinality	Medium
Clustering	No
Justification	To allow searching items by price range faster. Cardinality is medium

```
CREATE INDEX item_price ON item USING btree(price);
```

Index	IDX03
Related queries	SELECT08
Relation	wishlist
Attribute	user_id
Type	hash
Cardinality	Medium
Clustering	No
Justification	To allow for faster access to user wishlist. Cardinality is medium

```
CREATE INDEX wishlist_user_id ON wishlist USING hash(user_id);
```

Index	IDX04
Related queries	SELECT09
Relation	cart
Attribute	user_id
Type	hash
Cardinality	Medium
Clustering	No
Justification	To allow for faster access to user cart. Cardinality is medium

```
CREATE INDEX cart_user_id ON cart USING hash(user_id);
```

Index	IDX05
Related queries	SELECT12
Relation	review
Attribute	item_id
Type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	To allow for getting all the reviews of an item faster, clustering them. Cardinality is medium

```
CREATE INDEX item_review_idx ON review USING btree(item_id);
CLUSTER review using item_review_idx;
```

Index	IDX06
Related queries	SELECT15
Relation	advertisement
Attribute	begin_date
Type	B-tree
Cardinality	Medium
Clustering	No
Justification	To allow searching items by start date faster. Cardinality is medium

```
CREATE INDEX advertisement_start_date ON advertisement USING btree(begin_date);
```

Index	IDX07
Related queries	SELECT15
Relation	advertisement
Attribute	end_date
Type	B-tree
Cardinality	Medium
Clustering	No
Justification	To allow searching items by end date faster. Cardinality is medium

```
CREATE INDEX advertisement_end_date ON advertisement USING btree(end_date);
```

Index	IDX08
Related queries	SELECT18
Relation	purchase
Attribute	user_id
Type	hash
Cardinality	Medium
Clustering	No
Justification	To allow for faster access to user purchase. Cardinality is medium

```
CREATE INDEX purchase_user_id ON purchase USING hash(user_id);
```

2.2. Full-text Search Indices

Index	IDX09
Related queries	SELECT06
Relation	item
Attribute	search
Type	GIN
Clustering	No
Justification	To improve the performance of full text search on item. GIN because it's faster for lookups

```
CREATE INDEX item_search_index ON item USING GIN (search);
```

3. Triggers

Trigger	TRIGGER01
Description	When an item is removed, it is also removed from every user's cart and wishlist
`SQL code`	

```
DROP FUNCTION if exists remove_cart_and_wishlist CASCADE;
DROP TRIGGER if exists remove_archived_from_cart_and_wishlist ON item CASCADE;
CREATE FUNCTION remove_cart_and_wishlist() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF (OLD.is_archived <> NEW.is_archived) THEN
        DELETE FROM wishlist WHERE item_id = NEW.item_id;
    END IF;
    RETURN NEW;
END
$BODY$

LANGUAGE plpgsql;
CREATE TRIGGER remove_archived_from_cart_and_wishlist
AFTER UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE remove_cart_and_wishlist();
```

Trigger	TRIGGER02
Description	When an item out of stock gets stock, every user that has it in their wishlist gets a notification.
`SQL code`	

```
DROP FUNCTION if exists add_stock_notification CASCADE;
DROP TRIGGER if exists stock_notif ON item CASCADE;
CREATE FUNCTION add_stock_notification() RETURNS TRIGGER AS

$BODY$
BEGIN
    IF ( NEW.stock > OLD.stock) THEN
        IF(OLD.stock = 0) THEN
            INSERT INTO notification (user_id, discount_id, item_id, type)
            SELECT users.user_id, NULL, NEW.item_id, 'Stock'
            FROM users INNER JOIN wishlist using(user_id)
            WHERE wishlist.item_id = NEW.item_id;
        END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER stock_notif
AFTER UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE add_stock_notification();
```

Trigger	**TRIGGER03**
Description	When a review is added, the item's rating is recalculated.
`SQL code`	

```
DROP FUNCTION if exists update_score CASCADE;
DROP TRIGGER if exists score_on_review ON review CASCADE;
CREATE FUNCTION update_score() RETURNS TRIGGER AS

$BODY$
BEGIN
    UPDATE item
    SET score =
    (SELECT AVG(rating)
    FROM review
    WHERE review.item_id = NEW.item_id)
    WHERE item_id = NEW.item_id;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review
AFTER UPDATE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score();
```

Trigger	TRIGGER04
Description	When a review is updated, if the new score is different from the old score
SQL code	

```
DROP FUNCTION if exists update_score_change_review CASCADE;
DROP TRIGGER if exists score_on_review_change ON review CASCADE;
CREATE FUNCTION update_score_change_review () RETURNS TRIGGER AS

$BODY$
BEGIN
    IF (NEW.rating <> OLD.rating) THEN
        UPDATE item
        SET score =
        (SELECT AVG(rating)
        FROM review
        WHERE review.item_id = NEW.item_id)
        WHERE item_id = NEW.item_id;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review_change
AFTER UPDATE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score_change_review();
```

Trigger	TRIGGER05
Description	When a review is deleted, the item's score is updated. If there are no reviews left, the score gets set to 0.
SQL code	

```
DROP FUNCTION if exists update_score_delete CASCADE;
DROP TRIGGER if exists score_on_review_delete ON review CASCADE;
CREATE FUNCTION update_score_delete() RETURNS TRIGGER AS

$BODY$
BEGIN
    UPDATE item
    SET score =
    COALESCE((SELECT AVG(rating)
    FROM review
    WHERE review.item_id = OLD.item_id), 0)
    WHERE item_id = OLD.item_id;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review_delete
```

```
AFTER DELETE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score_delete();
```

Trigger	TRIGGER06
Description	When an item is added or updated, it's tsvector is updated for search
`SQL code`	

```
DROP TRIGGER IF EXISTS update_item_tsvector ON item;
DROP FUNCTION IF EXISTS update_item_tsvector() CASCADE;
CREATE FUNCTION update_item_tsvector() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF pg_trigger_depth() <=1 THEN
            update item
        set search = setweight(to_tsvector('english',coalesce(item.name,'')),
'A') ||
        setweight(to_tsvector('english',coalesce(item.description,'')), 'B')
        where new.item_id=item.item_id;

    END IF;
    RETURN NEW;

END

$BODY$
LANGUAGE plpgsql;
CREATE TRIGGER update_item_tsvector
AFTER INSERT OR UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE update_item_tsvector();
```

Trigger	TRIGGER07
Description	When a detail is added or updated to an item, its tsvector is updated for search
`SQL code`	

```
DROP TRIGGER IF EXISTS update_item_tsvector_detail ON item_detail;
DROP FUNCTION IF EXISTS update_item_tsvector_detail() CASCADE;
CREATE FUNCTION update_item_tsvector_detail() RETURNS TRIGGER AS
$BODY$
BEGIN
    update item

    set search = setweight(to_tsvector('english',coalesce(item.name,'')), 'A')
||
    setweight(to_tsvector('english',coalesce(item.description,'')), 'B') ||
setweight(to_tsvector('english',coalesce(s.detail_info,'')), 'C')
    from
    (
```

```
        select string_agg(detail_info, ' ') as detail_info
        from item_detail
        where new.item_id=item_detail.item_id
    ) as s
    where new.item_id=item.item_id;
    RETURN NEW;
END

$BODY$
LANGUAGE plpgsql;
CREATE TRIGGER update_item_tsvector_detail
AFTER INSERT OR UPDATE ON item_detail
FOR EACH ROW
EXECUTE PROCEDURE update_item_tsvector_detail();
```

Trigger	**TRIGGER08**
Description	When a user is banned, all their comments are deleted and the stock is added to the items
`SQL code`	

```
DROP FUNCTION if exists remove_banned_user_comments CASCADE;
DROP TRIGGER if exists remove_banned_user_comments ON ban CASCADE;


CREATE FUNCTION remove_banned_user_comments() RETURNS TRIGGER AS
$BODY$
BEGIN
    DELETE FROM review WHERE review.user_id = NEW.user_id;

    UPDATE item
    Set stock = item.stock + joined_cart.quantity
    FROM (cart INNER JOIN item using(item_id)) as joined_cart
    where user_id = NEW.user_id AND joined_cart.user_id = NEW.user_id AND
joined_cart.item_id = item.item_id;

    DELETE FROM cart
    WHERE user_id = NEW.user_id;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;


CREATE TRIGGER remove_banned_user_comments
AFTER INSERT ON ban
FOR EACH ROW
EXECUTE PROCEDURE remove_banned_user_comments();
```

Trigger	TRIGGER09
Description	If a user has an item already on the cart, instead of adding the same item again, its quantity is added.
SQL code	

```
DROP TRIGGER if exists check_if_already_on_cart ON cart CASCADE;
DROP FUNCTION if exists check_if_already_on_cart CASCADE;


CREATE FUNCTION check_if_already_on_cart() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF(EXISTS (SELECT * FROM cart WHERE (NEW.user_id = cart.user_id AND
NEW.item_id = cart.item_id)))
    THEN
        UPDATE cart
        SET quantity = quantity + NEW.quantity
        WHERE NEW.user_id = cart.user_id AND NEW.item_id = cart.item_id;
        RETURN NULL;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;


CREATE TRIGGER check_if_already_on_cart
BEFORE INSERT ON cart
FOR EACH ROW
EXECUTE PROCEDURE check_if_already_on_cart();
```

Trigger	RULE01
Description	When a user is deleted, instead of being deleted, most of their info that already isn't automatically set to null is set to null; deleted is set to true, is_admin is set to false and balance is set to 0. The user's photo and billing and shipping address table entry are delete. The user's cart is deleted and the items they had in the cart get the respective amout of stock added back.
SQL code	

```
DROP rule IF EXISTS users_delete_rule ON users CASCADE;

CREATE RULE users_delete_rule
AS ON DELETE TO users
DO INSTEAD (
    UPDATE item
    Set stock = item.stock + joined_cart.quantity
    FROM (cart INNER JOIN item using(item_id)) as joined_cart
    where user_id = OLD.user_id AND joined_cart.user_id = OLD.user_id AND
joined_cart.item_id = item.item_id;
    DELETE FROM cart
    WHERE user_id = OLD.user_id;
```

```
    UPDATE users
    SET first_name = null,
    last_name = null,
    username = null,
    email = null,
    password = null,
    deleted = true,
    is_admin = false,
    balance = 0
    WHERE OLD.user_id = user_id;
    DELETE FROM photo
    WHERE photo_id = OLD.img;
    DELETE FROM address
    WHERE address_id = OLD.shipping_address OR address_id = OLD.billing_address;
);
```

4. Transactions

> Transactions needed to assure the integrity of the data.

T01	Add to Cart, update stock
Justification	When adding an item to the cart, the stock must be decreased, but if more than one user adds the same item simultaneously, an error can occur and so the transaction is necessary. Repeatable Read isolation is used to avoid dirty and nonrepeatable reads, while allowing new rows to be inserted in items.
Isolation level	Repeatable Read

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

DO $$
BEGIN
    SELECT item_id
    FROM cart
    WHERE user_id = $user_id AND item_id = $item_id;

    IF NOT found THEN
        IF (
        SELECT stock
        FROM item
        WHERE item_id = $item_id) >= $quantity
        THEN
            UPDATE item
            SET stock = stock - $quantity
            WHERE item_id = $item_id;

            INSERT INTO cart
            VALUES($user_id, $item_id, now()::DATE, $quantity);

        END IF;
    ELSE
        IF (
            SELECT stock
```

```
            FROM item
            WHERE item_id = $item_id) >= $quantity
        THEN
            UPDATE item
            SET stock = stock - $quantity
            WHERE item_id = $item_id;

            UPDATE cart
            SET quantity = quantity + $quantity;

        END IF;
    END IF;
END
$$;

COMMIT;
```

T02	Balance and purchase on Checkout
Justification	When a user performs a checkout using money from their account balance, it is important to make sure that their balance and the cart are not updated externally during the operation, and so it is necessary to use a transaction. Also, the purchase records with the correct discounts should be added, should the discounts change during the operation. Serializable isolation is used to avoid dirty and nonrepeatable reads on the balance and cart tables while also making sure that all necessary rows, the items in the cart, are read (no phantoms).
Isolation level	Serializable

```
DROP FUNCTION if exists get_discount CASCADE;
CREATE FUNCTION get_discount(i INTEGER, d TIMESTAMP WITH TIME ZONE)
RETURNS INTEGER AS
$$
DECLARE item_discount INTEGER := 0;
BEGIN
    SELECT max(discount.percentage) INTO item_discount
    FROM apply_discount JOIN discount USING (discount_id)
    WHERE item_id = $1 AND begin_date <= $2 AND end_date >= $2;

    if(item_discount IS NULL) then
        RETURN 0;
    else
        return item_discount;
    end if;
END;
$$
LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
DO
$$
DECLARE sum_prices MONEY := 0::MONEY;
DECLARE purchase_ident INTEGER := 0;
```

```
BEGIN
        SELECT sum((price - price*get_discount(item_id, now())) * quantity) INTO
sum_prices
        FROM item JOIN cart USING (item_id)
        WHERE user_id = $user_id;

    IF (

        (SELECT balance
        FROM users
        WHERE user_id = $user_id)
        -
        (sum_prices)
        >= 0::MONEY
        )
        THEN

            UPDATE users
            SET balance = balance - sum_prices
            WHERE user_id = $user_id;

            INSERT INTO purchase(user_id,date) VALUES ($user_id, now())
RETURNING purchase_id INTO purchase_ident;

            INSERT INTO purchase_item (purchase_id, item_id, price, quantity)
                SELECT purchase_ident, item_id, (price-
price*get_discount(item_id, now())) * quantity, quantity
                FROM item JOIN cart USING (item_id)
                WHERE user_id = $user_id;

    END IF;
END
$$;

COMMIT;
```

Annex A. Complete SQL Code

4. SQL Code

A.1. Database schema

```
DROP TABLE IF EXISTS category_detail CASCADE;
DROP TABLE IF EXISTS item_detail CASCADE;
DROP TABLE IF EXISTS apply_discount CASCADE;
DROP TABLE IF EXISTS notification CASCADE;
DROP TABLE IF EXISTS discount CASCADE;
DROP TABLE IF EXISTS wishlist CASCADE;
DROP TABLE IF EXISTS cart CASCADE;
DROP TABLE IF EXISTS item_photo CASCADE;
DROP TABLE IF EXISTS advertisement CASCADE;
DROP TABLE IF EXISTS purchase_item CASCADE;
DROP TABLE IF EXISTS purchase CASCADE;
DROP TABLE IF EXISTS ban CASCADE;
DROP TABLE IF EXISTS review CASCADE;
DROP TABLE IF EXISTS item CASCADE;
```

```sql
DROP TABLE IF EXISTS details CASCADE;
DROP TABLE IF EXISTS category CASCADE;
DROP TABLE IF EXISTS users CASCADE;
DROP TABLE IF EXISTS address CASCADE;
DROP TABLE IF EXISTS photo CASCADE;
DROP TABLE IF EXISTS country CASCADE;

DROP TYPE IF EXISTS notificationType;

CREATE TYPE notificationType AS ENUM ('Stock','Discount');

CREATE TABLE country (
    country_id SERIAL PRIMARY KEY,
    name text NOT NULL CONSTRAINT country_name_uk UNIQUE
);


CREATE TABLE photo (
    photo_id SERIAL PRIMARY KEY,
    path text NOT NULL
);

CREATE TABLE address (
    address_id SERIAL PRIMARY KEY,
    city text NOT NULL,
    street text NOT NULL,
    zip_code text NOT NULL,
    country_id INTEGER REFERENCES country(country_id) ON UPDATE CASCADE
);

CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
    username text CONSTRAINT username_uk UNIQUE,
    email text CONSTRAINT user_email_uk UNIQUE,
    first_name text,
    last_name text,
    password text,
    deleted BOOLEAN DEFAULT FALSE,
    is_admin BOOLEAN DEFAULT FALSE,
    balance MONEY DEFAULT 0,
    img INTEGER REFERENCES photo(photo_id) ON UPDATE CASCADE,
    billing_address INTEGER REFERENCES address(address_id) ON UPDATE CASCADE,
    shipping_address INTEGER REFERENCES address(address_id) ON UPDATE CASCADE
);

CREATE TABLE category (
    category_id SERIAL PRIMARY KEY,
    name text NOT NULL UNIQUE
);

CREATE TABLE details (
    detail_id SERIAL PRIMARY KEY,
    name text NOT NULL UNIQUE
);

CREATE TABLE item (
    item_id SERIAL PRIMARY KEY,
    name text NOT NULL,
```

```
    stock INTEGER NOT NULL CONSTRAINT pos_stock CHECK (stock >= 0),
    brief_description text,
    description text NOT NULL,
    price MONEY NOT NULL CONSTRAINT pos_price CHECK (price >= 0::MONEY),
    is_archived BOOLEAN NOT NULL DEFAULT false,
    category_id INTEGER REFERENCES category (category_id) ON UPDATE CASCADE,
    score REAL NOT NULL CONSTRAINT rating_ck CHECK (((score >= 0) AND (score <=
5))),
    search tsvector
);

CREATE TABLE review (
    review_id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(user_id) ON UPDATE CASCADE,
    item_id INTEGER REFERENCES item(item_id) ON UPDATE CASCADE,
    comment_text text,
    "date" TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    rating INTEGER NOT NULL CONSTRAINT rating_ck CHECK (((rating > 0) AND
(rating <= 5)))
);

CREATE TABLE ban (
    admin_id INTEGER NOT NULL REFERENCES users(user_id) ON UPDATE CASCADE,
    user_id INTEGER NOT NULL REFERENCES users(user_id) ON UPDATE CASCADE,
    "date" TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    reason text NOT NULL,
    PRIMARY KEY (admin_id, user_id)
);

CREATE TABLE purchase (
    purchase_id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(user_id) ON UPDATE CASCADE,
    "date" TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL
);

CREATE TABLE purchase_item (
    purchase_id INTEGER NOT NULL REFERENCES purchase (purchase_id) ON UPDATE
CASCADE,
    item_id INTEGER NOT NULL REFERENCES item (item_id) ON UPDATE CASCADE,
    price MONEY NOT NULL,
    quantity INTEGER NOT NULL CONSTRAINT quantity_more_zero CHECK (quantity >
0),
    PRIMARY KEY (purchase_id, item_id)
);

CREATE TABLE advertisement (
    advertisement_id INTEGER PRIMARY KEY,
    title text NOT NULL,
    begin_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    end_date TIMESTAMP WITH TIME zone NOT NULL,
    photo_id INTEGER REFERENCES photo (photo_id) ON UPDATE CASCADE,
    CONSTRAINT ad_dates_ck CHECK (begin_date < end_date)
);

CREATE TABLE item_photo (
    photo_id INTEGER NOT NULL REFERENCES photo (photo_id) ON UPDATE CASCADE
PRIMARY KEY,
    item_id INTEGER NOT NULL REFERENCES item (item_id) ON UPDATE CASCADE
```

```
);


CREATE TABLE cart (
    user_id INTEGER REFERENCES users(user_id) ON UPDATE CASCADE,
    item_id INTEGER NOT NULL REFERENCES item (item_id) ON UPDATE CASCADE,
    add_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    quantity INTEGER NOT NULL CONSTRAINT quantity_more_zero CHECK (quantity >
0),
    PRIMARY KEY (user_id, item_id)
);


CREATE TABLE wishlist (
    user_id INTEGER REFERENCES users(user_id) ON UPDATE CASCADE,
    item_id INTEGER NOT NULL REFERENCES item(item_id) ON UPDATE CASCADE,
    add_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    PRIMARY KEY (user_id, item_id)
);

CREATE TABLE discount (
    discount_id SERIAL PRIMARY KEY,
    percentage INTEGER NOT NULL CONSTRAINT valid_percentage CHECK (((percentage
> 0) AND (percentage <= 100))),
    begin_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    end_date TIMESTAMP WITH TIME zone NOT NULL,
    CONSTRAINT ad_dates_ck CHECK (begin_date < end_date)
);

CREATE TABLE notification (
    user_id INTEGER REFERENCES users(user_id) ON UPDATE CASCADE,
    discount_id INTEGER REFERENCES discount (discount_id) ON UPDATE CASCADE,
    notification_id SERIAL PRIMARY KEY,
    item_id INTEGER NOT NULL REFERENCES item(item_id) ON UPDATE CASCADE,
    type notificationType,
    is_seen Boolean DEFAULT False
);


CREATE TABLE apply_discount (
    item_id INTEGER NOT NULL REFERENCES item (item_id) ON UPDATE CASCADE,
    discount_id INTEGER NOT NULL REFERENCES discount (discount_id) ON UPDATE
CASCADE,
    PRIMARY KEY (item_id, discount_id)
);

CREATE TABLE item_detail (
    item_id INTEGER NOT NULL REFERENCES item (item_id) ON UPDATE CASCADE,
    detail_id INTEGER NOT NULL REFERENCES details (detail_id) ON UPDATE CASCADE,
    detail_info text NOT NULL,
    PRIMARY KEY (detail_id, item_id)
);

CREATE TABLE category_detail (
    category_id INTEGER NOT NULL REFERENCES category (category_id) ON UPDATE
CASCADE,
    detail_id INTEGER NOT NULL REFERENCES details (detail_id) ON UPDATE CASCADE,
```

```
    PRIMARY KEY (category_id, detail_id)
);


-- Indexes

DROP INDEX IF EXISTS item_detail_itemID CASCADE;

CREATE INDEX item_detail_itemID ON item_detail USING btree(item_id);
CLUSTER item_detail using item_detail_itemID;


DROP INDEX IF EXISTS item_price CASCADE;
CREATE INDEX item_price ON item USING btree(price);

DROP INDEX IF EXISTS wishlist_user_id CASCADE;
CREATE INDEX wishlist_user_id ON wishlist USING hash(user_id);

DROP INDEX IF EXISTS cart_user_id CASCADE;
CREATE INDEX cart_user_id ON cart USING hash(user_id);

DROP INDEX IF EXISTS item_review_idx CASCADE;
CREATE INDEX item_review_idx ON review USING btree(item_id);
CLUSTER review using item_review_idx;


DROP INDEX IF EXISTS advertisement_start_date CASCADE;
CREATE INDEX advertisement_start_date ON advertisement USING btree(begin_date);

DROP INDEX IF EXISTS advertisement_end_date CASCADE;
CREATE INDEX advertisement_end_date ON advertisement USING btree(end_date);

DROP INDEX IF EXISTS purchase_user_id CASCADE;
CREATE INDEX purchase_user_id ON purchase USING hash(user_id);

DROP INDEX IF EXISTS item_search_index CASCADE;
CREATE INDEX item_search_index ON item USING GIN (search);

-- Triggers


-- Trigger 1
DROP FUNCTION if exists remove_cart_and_wishlist CASCADE;
DROP TRIGGER if exists remove_archived_from_cart_and_wishlist ON item CASCADE;
CREATE FUNCTION remove_cart_and_wishlist() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF (OLD.is_archived <> NEW.is_archived) THEN
        DELETE FROM wishlist WHERE item_id = NEW.item_id;
    END IF;
    RETURN NEW;
END
$BODY$

LANGUAGE plpgsql;
CREATE TRIGGER remove_archived_from_cart_and_wishlist
AFTER UPDATE ON item
FOR EACH ROW
```

```
EXECUTE PROCEDURE remove_cart_and_wishlist();


-- Trigger 2
DROP FUNCTION if exists add_stock_notification CASCADE;
DROP TRIGGER if exists stock_notif ON item CASCADE;
CREATE FUNCTION add_stock_notification() RETURNS TRIGGER AS

$BODY$
BEGIN
    IF ( NEW.stock > OLD.stock) THEN
        IF(OLD.stock = 0) THEN
            INSERT INTO notification (user_id, discount_id, item_id, type)
            SELECT users.user_id, NULL, NEW.item_id, 'Stock'
            FROM users INNER JOIN wishlist using(user_id)
            WHERE wishlist.item_id = NEW.item_id;
        END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER stock_notif
AFTER UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE add_stock_notification();


-- Trigger 3
DROP FUNCTION if exists update_score CASCADE;
DROP TRIGGER if exists score_on_review ON review CASCADE;
CREATE FUNCTION update_score() RETURNS TRIGGER AS

$BODY$
BEGIN
    UPDATE item
    SET score =
    (SELECT AVG(rating)
    FROM review
    WHERE review.item_id = NEW.item_id)
    WHERE item_id = NEW.item_id;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review
AFTER UPDATE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score();


-- Trigger 4
DROP FUNCTION if exists update_score_change_review CASCADE;
DROP TRIGGER if exists score_on_review_change ON review CASCADE;
CREATE FUNCTION update_score_change_review () RETURNS TRIGGER AS
```

```
$BODY$
BEGIN
    IF (NEW.rating <> OLD.rating) THEN
        UPDATE item
        SET score =
        (SELECT AVG(rating)
        FROM review
        WHERE review.item_id = NEW.item_id)
        WHERE item_id = NEW.item_id;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review_change
AFTER UPDATE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score_change_review();


--Trigger 5
DROP FUNCTION if exists update_score_delete CASCADE;
DROP TRIGGER if exists score_on_review_delete ON review CASCADE;
CREATE FUNCTION update_score_delete() RETURNS TRIGGER AS

$BODY$
BEGIN
    UPDATE item
    SET score =
    COALESCE((SELECT AVG(rating)
    FROM review
    WHERE review.item_id = OLD.item_id), 0)
    WHERE item_id = OLD.item_id;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER score_on_review_delete
AFTER DELETE ON review
FOR EACH ROW
EXECUTE PROCEDURE update_score_delete();


--Trigger 6
DROP TRIGGER IF EXISTS update_item_tsvector ON item;
DROP FUNCTION IF EXISTS update_item_tsvector() CASCADE;
CREATE FUNCTION update_item_tsvector() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF pg_trigger_depth() <=1 THEN
            update item
        set search = setweight(to_tsvector('english',coalesce(item.name,'')),
'A') ||
        setweight(to_tsvector('english',coalesce(item.description,'')), 'B')
        where new.item_id=item.item_id;
    END IF;
```

```
    RETURN NEW;

END

$BODY$
LANGUAGE plpgsql;
CREATE TRIGGER update_item_tsvector
AFTER INSERT OR UPDATE ON item
FOR EACH ROW
EXECUTE PROCEDURE update_item_tsvector();


--Trigger 7
DROP TRIGGER IF EXISTS update_item_tsvector_detail ON item_detail;
DROP FUNCTION IF EXISTS update_item_tsvector_detail() CASCADE;
CREATE FUNCTION update_item_tsvector_detail() RETURNS TRIGGER AS
$BODY$
BEGIN
    update item

    set search = setweight(to_tsvector('english',coalesce(item.name,'')), 'A')
||
    setweight(to_tsvector('english',coalesce(item.description,'')), 'B') ||
setweight(to_tsvector('english',coalesce(s.detail_info,'')), 'C')
    from
    (

        select string_agg(detail_info, ' ') as detail_info
        from item_detail
        where new.item_id=item_detail.item_id
    ) as s
    where new.item_id=item.item_id;
    RETURN NEW;
END
$BODY$


LANGUAGE plpgsql;
CREATE TRIGGER update_item_tsvector_detail
AFTER INSERT OR UPDATE ON item_detail
FOR EACH ROW
EXECUTE PROCEDURE update_item_tsvector_detail();



--Trigger 8
DROP FUNCTION if exists remove_banned_user_comments CASCADE;
DROP TRIGGER if exists remove_banned_user_comments ON ban CASCADE;
CREATE FUNCTION remove_banned_user_comments() RETURNS TRIGGER AS
$BODY$
BEGIN
    DELETE FROM review WHERE review.user_id = NEW.user_id;

    UPDATE item
    Set stock = item.stock + joined_cart.quantity
    FROM (cart INNER JOIN item using(item_id)) as joined_cart
    where user_id = NEW.user_id AND joined_cart.user_id = NEW.user_id AND
joined_cart.item_id = item.item_id;
```

```
    DELETE FROM cart
    WHERE user_id = NEW.user_id;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER remove_banned_user_comments
AFTER INSERT ON ban
FOR EACH ROW
EXECUTE PROCEDURE remove_banned_user_comments();


--Trigger 9
DROP TRIGGER if exists check_if_already_on_cart ON cart CASCADE;
DROP FUNCTION if exists check_if_already_on_cart CASCADE;

CREATE FUNCTION check_if_already_on_cart() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF(EXISTS (SELECT * FROM cart WHERE (NEW.user_id = cart.user_id AND
NEW.item_id = cart.item_id)))
    THEN
        UPDATE cart
        SET quantity = quantity + NEW.quantity
        WHERE NEW.user_id = cart.user_id AND NEW.item_id = cart.item_id;
        RETURN NULL;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER check_if_already_on_cart
BEFORE INSERT ON cart
FOR EACH ROW
EXECUTE PROCEDURE check_if_already_on_cart();


--Rule 1
DROP rule IF EXISTS users_delete_rule ON users CASCADE;
CREATE RULE users_delete_rule
AS ON DELETE TO users
DO INSTEAD (
    UPDATE item
    Set stock = item.stock + joined_cart.quantity
    FROM (cart INNER JOIN item using(item_id)) as joined_cart
    where user_id = OLD.user_id AND joined_cart.user_id = OLD.user_id AND
joined_cart.item_id = item.item_id;
    DELETE FROM cart
    WHERE user_id = OLD.user_id;
    UPDATE users
    SET first_name = null,
    last_name = null,
    username = null,
    email = null,
    password = null,
```

```
    deleted = true,
    is_admin = false,
    balance = 0
    WHERE OLD.user_id = user_id;
    DELETE FROM photo
    WHERE photo_id = OLD.img;
    DELETE FROM address
    WHERE address_id = OLD.shipping_address OR address_id = OLD.billing_address;
);


--Transactions

--Transaction 1

CREATE OR REPLACE PROCEDURE remove_stock(userID INTEGER, itemID INTEGER,
quantityBought INTEGER)
LANGUAGE plpgsql AS $$
DECLARE
BEGIN
    SELECT item_id
    FROM cart
    WHERE user_id = userID AND item_id = itemID;

    IF NOT found THEN
        IF (
        SELECT stock
        FROM item
        WHERE item_id = itemID) >= quantityBought
        THEN
            UPDATE item
            SET stock = stock - quantityBought
            WHERE item_id = itemID;

            INSERT INTO cart
            VALUES(userID, itemID, now()::DATE, quantityBought);

        END IF;
    ELSE
        IF (
            SELECT stock
            FROM item
            WHERE item_id = itemID) >= quantityBought
        THEN
            UPDATE item
            SET stock = stock - quantityBought
            WHERE item_id = itemID;

            UPDATE cart
            SET quantity = quantity + quantityBought;

        END IF;
    END IF;
END
$$;

COMMIT;
```

```
--Transaction 2

DROP FUNCTION if exists get_discount CASCADE;
CREATE FUNCTION get_discount(i INTEGER, d TIMESTAMP WITH TIME ZONE)
RETURNS INTEGER AS
$$
DECLARE item_discount INTEGER := 0;
BEGIN
    SELECT max(discount.percentage) INTO item_discount
    FROM apply_discount JOIN discount USING (discount_id)
    WHERE item_id = $1 AND begin_date <= $2 AND end_date >= $2;

    if(item_discount IS NULL) then
        RETURN 0;
    else
        return item_discount;
    end if;
END;
$$
LANGUAGE plpgsql;

CREATE OR REPLACE PROCEDURE discounts(userID INTEGER)
LANGUAGE plpgsql AS $$
DECLARE
sum_prices MONEY := 0::MONEY;
purchase_ident INTEGER := 0;
BEGIN
SELECT sum((price - price*get_discount(item_id, now())) * quantity) INTO
sum_prices
        FROM item JOIN cart USING (item_id)
        WHERE cart.user_id = userID;

    IF (

        (SELECT balance
        FROM users
        WHERE user_id = userID)
        -
        (sum_prices)
        >= 0::MONEY
        )
        THEN

            UPDATE users
            SET balance = balance - sum_prices
            WHERE user_id = userID;

            INSERT INTO purchase(user_id,date) VALUES (userID, now()) RETURNING
purchase_id INTO purchase_ident;

            INSERT INTO purchase_item (purchase_id, item_id, price, quantity)
                SELECT purchase_ident, item_id, (price-
price*get_discount(item_id, now())) * quantity, quantity
                FROM item JOIN cart USING (item_id)
                WHERE user_id = userID;

    END IF;
END
```

```
$$;
commit;
```

A.2. Database population

```
INSERT INTO category (category_id, name) VALUES (1, 'Computers');
INSERT INTO category (category_id, name) VALUES (2, 'Books');
INSERT INTO category (category_id, name) VALUES (3, 'Eletrodomestics');
INSERT INTO category (category_id, name) VALUES (4, 'Video Games');
INSERT INTO category (category_id, name) VALUES (5, 'Smartphones');
INSERT INTO category (category_id, name) VALUES (6, 'Headphones');
INSERT INTO category (category_id, name) VALUES (7, 'Music CDs, Vinil');
INSERT INTO category (category_id, name) VALUES (8, 'School Material');
INSERT INTO category (category_id, name) VALUES (9, 'Home Theatres');
INSERT INTO category (category_id, name) VALUES (10, 'Televisions');
INSERT INTO category (category_id, name) VALUES (11, 'Movies');



INSERT INTO country (country_id, name) VALUES (1, 'Spain');
INSERT INTO country (country_id, name) VALUES (2, 'Bulgaria');
INSERT INTO country (country_id, name) VALUES (3, 'Western Sahara');
INSERT INTO country (country_id, name) VALUES (4, 'Barbados');
INSERT INTO country (country_id, name) VALUES (5, 'Sri Lanka');
INSERT INTO country (country_id, name) VALUES (6, 'Gibraltar');
INSERT INTO country (country_id, name) VALUES (7, 'Svalbard and Jan Mayen
Islands');
INSERT INTO country (country_id, name) VALUES (8, 'Antigua and Barbuda');
INSERT INTO country (country_id, name) VALUES (9, 'Burundi');
INSERT INTO country (country_id, name) VALUES (10, 'Norfolk Island');
INSERT INTO country (country_id, name) VALUES (11, 'Costa Rica');
INSERT INTO country (country_id, name) VALUES (12, 'Czech Republic');
INSERT INTO country (country_id, name) VALUES (13, 'Mozambique');
INSERT INTO country (country_id, name) VALUES (14, 'Ethiopia');
INSERT INTO country (country_id, name) VALUES (15, 'Micronesia');
INSERT INTO country (country_id, name) VALUES (16, 'Tajikistan');
INSERT INTO country (country_id, name) VALUES (17, 'United States');
INSERT INTO country (country_id, name) VALUES (18, 'Peru');
INSERT INTO country (country_id, name) VALUES (19, 'Portugal');
INSERT INTO country (country_id, name) VALUES (20, 'Tokelau');



INSERT INTO details (detail_id, name) VALUES (1, 'Processor');
INSERT INTO details (detail_id, name) VALUES (2, 'RAM');
INSERT INTO details (detail_id, name) VALUES (3, 'Storage');
INSERT INTO details (detail_id, name) VALUES (4, 'Graphics');
INSERT INTO details (detail_id, name) VALUES (5, 'Release Date');
INSERT INTO details (detail_id, name) VALUES (6, 'Age restriction');
INSERT INTO details (detail_id, name) VALUES (7, 'Developer');
INSERT INTO details (detail_id, name) VALUES (8, 'Camera');
INSERT INTO details (detail_id, name) VALUES (9, 'Screen Size');
INSERT INTO details (detail_id, name) VALUES (10, 'Screen Resolution');
INSERT INTO details (detail_id, name) VALUES (11, 'InfraRed');
INSERT INTO details (detail_id, name) VALUES (12, 'Bluetooth');
INSERT INTO details (detail_id, name) VALUES (13, 'Wireless');
INSERT INTO details (detail_id, name) VALUES (14, 'Battery');
```

```
INSERT INTO details (detail_id, name) VALUES (15, 'Author');
INSERT INTO details (detail_id, name) VALUES (16, 'Edition');
INSERT INTO details (detail_id, name) VALUES (17, 'Power Consumption');
INSERT INTO details (detail_id, name) VALUES (18, 'Guarantee');
INSERT INTO details (detail_id, name) VALUES (19, 'Dimensions');
INSERT INTO details (detail_id, name) VALUES (20, 'Energy Label');
INSERT INTO details (detail_id, name) VALUES (21, 'Weight');
INSERT INTO details (detail_id, name) VALUES (22, 'Genre');
INSERT INTO details (detail_id, name) VALUES (23, 'Disc Format');
INSERT INTO details (detail_id, name) VALUES (24, 'Sound');


INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (1,
46, '2020-08-11 08:47:16', '2020-08-16 03:45:26');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (2,
26, '2020-08-31 03:55:46', '2021-02-22 17:58:47');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (3,
49, '2020-04-10 15:30:47', '2020-11-15 02:45:01');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (4,
52, '2020-03-21 01:06:07', '2020-11-26 16:28:20');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (5,
10, '2020-03-02 19:34:54', '2020-05-03 07:55:31');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (6,
28, '2020-04-08 21:26:30', '2020-08-24 16:45:42');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (7,
54, '2020-11-28 22:47:23', '2021-02-07 15:13:13');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (8,
65, '2020-05-26 19:10:39', '2021-06-22 10:11:30');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (9,
52, '2020-12-31 09:01:14', '2021-01-10 05:48:03');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (10,
82, '2020-12-30 12:56:22', '2021-01-02 10:19:08');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (11,
76, '2020-02-02 19:24:15', '2021-11-10 19:43:29');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (12,
1, '2020-04-01 13:01:47', '2020-10-01 13:04:25');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (13,
45, '2020-05-27 16:12:42', '2020-11-18 10:38:14');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (14,
17, '2020-05-10 20:51:33', '2020-09-14 18:14:25');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (15,
18, '2021-03-06 18:44:44', '2021-08-06 11:52:43');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (16,
34, '2020-02-25 07:00:38', '2020-12-29 08:28:27');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (17,
23, '2020-02-26 15:01:32', '2020-05-11 06:45:20');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (18,
15, '2020-12-25 11:36:42', '2021-04-12 05:23:22');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (19,
60, '2020-03-25 19:19:40', '2021-03-21 01:12:51');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (20,
21, '2020-04-04 17:13:35', '2021-01-22 18:43:04');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (21,
72, '2020-02-16 22:55:32', '2020-04-22 15:56:40');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (22,
54, '2021-01-04 11:12:39', '2021-08-18 01:10:42');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (23,
35, '2020-08-04 02:20:07', '2021-02-07 15:29:23');
```

```
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (24, 26, '2020-11-03 11:30:28', '2021-01-13 00:58:57');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (25, 19, '2020-04-03 19:52:58', '2020-09-27 13:37:19');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (26, 54, '2020-08-11 22:01:24', '2020-09-16 13:20:04');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (27, 15, '2020-03-08 18:11:59', '2020-05-26 02:43:08');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (28, 77, '2021-03-15 21:17:29', '2021-04-15 01:36:31');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (29, 39, '2020-10-28 14:20:53', '2020-11-14 06:16:17');
INSERT INTO discount (discount_id, percentage, begin_date, end_date) VALUES (30, 88, '2021-02-23 12:32:37', '2021-08-24 14:46:22');



INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (1, 'ASUS laptop MSXKD-231', 25, 'Cutting-edge laptop for all your needs', 'With the most recent techonology in the market, this laptop will blow you away with its tremendous power, allowing for a smooth experience even in gaming. Designed with the first 300Hz screen on a laptop, it is as powerful as it is quiet.', 2500, False, 1, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (2, 'Lenovo Thinkpad', 92, 'Small and quiet, useful on the go', '', 650, False, 1, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (3, 'My Hero Academia - Book 5', 86, NULL, 'The final stages of the U.A. High sports festival promise to be explosive as Uraraka takes on Bakugo in a head to head match! Bakugo never gives anyone a break, and the crowd holds its breath as the battle begins. The finals will push the students of Class 1-A to their limits and beyond!', 10, False, 2, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (4, 'The Intelligent Investor', 37, 'The Definitive Book on Value Investing', 'Presents the philosophy of "value investing", which helps protect investors against the areas of possible substantial error and teaches them to develop long-term strategies with which they will be comfortable down the road. This book enables you to make the right decisions to protect your investments and make them a success.', 19.60, False, 2, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (5, 'Robot Mi Vacuum Mop Pro', 49, 'Robot Vacuum', '', 349.99, True, 3, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (6, 'Microwave Beltax BMO-1120 - Silver', 59, NULL, '', 79.99, False, 3, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price, is_archived, category_id, score, search) VALUES (7, 'Resident Evil Village - PS4', 68, NULL, 'First-person action - You will take on the role of Ethan Winters and experience each battle at close range and terrifying chase from an extremely close perspective.\nFamiliar faces and new enemies - As a rule, Chris Redfield has been a hero in the Resident Evil series, but his appearance in Resident Evil Village appears to be involved in sinister reasons. A series of new adversaries who inhabit the enigmatic village will ruthlessly pursue Ethan, making his every move difficult, as he tries to understand the new nightmare in which he finds himself involved.', 69.99, False, 4, 0, '');
```

```
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (8, 'Minecraft Nintendo Switch
Edition', 79, NULL, 'The phenomenal and unforgettable full Minecraft experience,
now on the Nintendo Switch so you can play anywhere at anytime!', 29.99, False,
4, 0, NULL);
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (9, 'iPhone 12 Pro Max', 22,
'Most recent iPhone from Apple', 'High quality device designed at Apple for the
most refined and excellent smartphone experience', 1629.99, False, 5, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (10, 'Smartphone SAMSUNG Galaxy
Fold', 78, NULL, 'This new smartphone from the famous Samsung company is the
prime example of modern smartphone technology, with a fully working screen that
folds!', 6.8, False, 5, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (11, 'Headphones Bluetooth Sony
WH-1000XM3', 30, 'Wireless bluetooth headphones', '', 379.99, False, 6, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (12, 'Medicine at Midnight - LP
Orange Vinil', 84, 'Vinil exclusive from Foo Figthers', '', 21.99, False, 7, 0,
'');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (13, 'Minions Pencil set', 50,
NULL, 'Draw and paint with these pencils, from your cute and fun Minions!',
5.28, False, 8, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (14, 'Frozen Pencil set', 57,
NULL, 'Draw the most inspiring art with these pencils, inspired in your
favourite princesses from Frozen!', 6.80, False, 8, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (15, 'Soundbar Bluetooth Signa
S2', 29, NULL, '', 249.99, False, 9, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (16, 'Smart TV Samsung UHD 4K
55AU7105', 68, NULL, 'This new TV from Samsung will blow you away with its
incredible screen size, resolution and beautiful colors', 699.99, False, 10, 0,
'');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (17, 'Smart TV Samsung Neo QLED
8K 65QN800A', 36, NULL, '8K is here and this TV proves it. Images have never
been shaper and colorful than with this new Samsung TV with the most cutting edge
of this time', 3520.30, False, 10, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (18, 'Fifty Shades Freed', 55,
NULL, 'The most recent installment on the Fifty Shades series now with an
extended DVD version, with scenes never before scene on theatre displays', 7.99,
False, 11, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (19, 'How to Train your Dragon
3', 8, NULL, '', 13.99, False, 11, 0, '');
INSERT INTO item (item_id, name, stock, brief_description, description, price,
is_archived, category_id, score, search) VALUES (20, 'Robocop Trilogy Blu-ray',
50, NULL, 'The whole Robocop trilogy, now on Blu-ray with new scenes and
enhanced image quality', 10.00, False, 11, 0, '');
```

```
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (1, 1, 'Intel
i7-10700H');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (1, 2, '16
GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (1, 3, '1TB
HDD, 250GB SSD');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (1, 4, 'Nvidia
RTX3060 6GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (1, 9,
'15,6"');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (2, 1, 'Intel
i5-9700H');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (2, 2, '8 GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (2, 3, '500GB
SSD');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (2, 4, 'Nvidia
MX250');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (2, 9, '14"');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (3, 15, 'Kohei
Horikoshi');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (3, 16, '2');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (3, 22,
'Action');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (4, 15,
'Benjamin Graham');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (4, 16, '1');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (4, 22,
'Business');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (5, 17, '10W');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (5, 18, '1
year');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (5, 19,
'250x250x30 mm');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (5, 21, '3kg');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (6, 17,
'305W');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (6, 18, '5
years');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (6, 19,
'50x50x40 cm');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (6, 20, 'A++');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (6, 21,
'9.5kg');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (7, 5,
'25/04/2021');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (7, 6, '18+');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (7, 7,
'Capcom');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (8, 5,
'14/03/2021');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (8, 6, '3+');
```

```
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (8, 7,
'Mojang');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 2, '12GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 3,
'128GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 8, '12MP +
12MP + 12MP');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 9, '6,7"');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 10,
'2778x1284 px');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 11, 'Yes');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 12, 'Yes');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (9, 14, '5000
mAh');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 1,
'Snapdragon 885');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 2,
'12GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 3,
'128GB');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 8, '12MP +
16MP + 12MP');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 9,
'7,3"');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 10,
'QXGA+');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 11,
'Yes');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 12,
'Yes');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (10, 14, '4380
mAh');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (11, 12,
'Yes');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (11, 14,
'Rechargable 3000 mAh');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (11, 21,
'255g');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (11, 24, '104,5
dB');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (12, 5,
'February 2021');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (12, 15, 'Foo
Fighters');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (12, 23,
'Vinil');

--INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (13, 19, '');
--INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (14, 19, '');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (15, 18, '1
year');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (15, 19, '89,99
x 8,18 x 5,46 cm');
```

```
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (16, 9, '50"');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (16, 10, '4K
3840x2160');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (16, 17,
'20W');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (16, 18, '2
years');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (17, 9, '65"');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (17, 10, '8K
7680x4320');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (17, 17,
'70W');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (17, 18, '2
years');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (17, 21, '30.8
kg');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (18, 6, '18+');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (18, 23,
'DVD');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (19, 5,
'2019');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (19, 6, '7+');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (19, 23,
'DVD');

INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (20, 6, '12+');
INSERT INTO item_detail (item_id, detail_id, detail_info) VALUES (20, 23, 'Blu-
ray');



INSERT INTO photo (photo_id, path) VALUES (1,
'http://dummyimage.com/219x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (2,
'http://dummyimage.com/162x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (3,
'http://dummyimage.com/235x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (4,
'http://dummyimage.com/164x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (5,
'http://dummyimage.com/214x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (6,
'http://dummyimage.com/186x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (7,
'http://dummyimage.com/131x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (8,
'http://dummyimage.com/195x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (9,
'http://dummyimage.com/163x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (10,
'http://dummyimage.com/231x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (11,
'http://dummyimage.com/195x100.png/dddddd/000000');
```

```
INSERT INTO photo (photo_id, path) VALUES (12,
'http://dummyimage.com/122x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (13,
'http://dummyimage.com/143x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (14,
'http://dummyimage.com/139x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (15,
'http://dummyimage.com/104x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (16,
'http://dummyimage.com/232x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (17,
'http://dummyimage.com/127x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (18,
'http://dummyimage.com/204x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (19,
'http://dummyimage.com/151x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (20,
'http://dummyimage.com/182x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (21,
'http://dummyimage.com/207x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (22,
'http://dummyimage.com/158x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (23,
'http://dummyimage.com/207x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (24,
'http://dummyimage.com/222x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (25,
'http://dummyimage.com/174x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (26,
'http://dummyimage.com/217x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (27,
'http://dummyimage.com/224x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (28,
'http://dummyimage.com/189x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (29,
'http://dummyimage.com/116x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (30,
'http://dummyimage.com/180x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (31,
'http://dummyimage.com/129x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (32,
'http://dummyimage.com/107x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (33,
'http://dummyimage.com/156x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (34,
'http://dummyimage.com/186x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (35,
'http://dummyimage.com/242x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (36,
'http://dummyimage.com/156x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (37,
'http://dummyimage.com/161x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (38,
'http://dummyimage.com/175x100.png/dddddd/000000');
INSERT INTO photo (photo_id, path) VALUES (39,
'http://dummyimage.com/212x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (40,
'http://dummyimage.com/118x100.png/cc0000/ffffff');
```

```
INSERT INTO photo (photo_id, path) VALUES (41,
'http://dummyimage.com/227x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (42,
'http://dummyimage.com/181x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (43,
'http://dummyimage.com/182x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (44,
'http://dummyimage.com/219x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (45,
'http://dummyimage.com/185x100.png/5fa2dd/ffffff');
INSERT INTO photo (photo_id, path) VALUES (46,
'http://dummyimage.com/153x100.png/cc0000/ffffff');
INSERT INTO photo (photo_id, path) VALUES (47,
'http://dummyimage.com/184x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (48,
'http://dummyimage.com/183x100.png/ff4444/ffffff');
INSERT INTO photo (photo_id, path) VALUES (49,
'http://dummyimage.com/108x100.png/ff4444/ffffff');



INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (1,
'Davao', 'Seventh Ave', '31596', 10);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (2,
'Odessa', 'Sixth Ave', '35704', 10);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (3,
'Weifang', 'Richmond Road', '62326', 10);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (4,
'Haicheng', 'Washington Street', '32984', 11);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (5,
'HARARE', 'Queens Street', '72190', 9);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (6,
'Quezon City', 'MapleLane', '32899', 11);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (7,
'Birmingham', 'Oak Way', '49361', 18);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (8,
'MANAGUA', 'SouthDrive', '81430', 18);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (9,
'NAIROBI', 'Eighth Drive', '55504', 18);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (10,
'Samara', 'WindsorRoad', '49651', 19);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (11,
'Huzhou', 'QueensRoad', '79251', 19);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (12,
'Pikine-Guediawaye', 'WesRoad', '11969', 19);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (13,
'Nanning', 'Pine Way', '09105', 20);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (14,
'Lagos', 'NinthRoad', '78185', 20);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (15,
'RABAT', 'SeventhRoad', '90241', 20);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (16,
'Qinzhou', 'Windsor Lane', '13276', 3);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (17,
'C◆rdoba', 'Second Way', '10108', 3);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (18,
'Havanna', 'MapleRoad', '11835', 4);
```

```
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (19,
'Fortaleza', 'WesLane', '28129', 4);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (20,
'Qiqihaer', 'NorthRoad', '63194', 4);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (21,
'Shiraz', 'GrangeAve', '21765', 11);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (22,
'HANOI', 'SecondDrive', '87453', 14);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (23,
'Changchun', 'York Ave', '85201', 14);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (24,
'HANOI', 'SouthWay', '79517', 3);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (25,
'Sendai', 'MillLane', '20998', 8);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (26,
'Tianjin', 'Kings Way', '26333', 8);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (27,
'Yulin', 'Oak Drive', '31168', 8);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (28,
'Changsha', 'Eighth Road', '24709', 9);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (29,
'SANTIAGO', 'SouthDrive', '55296', 12);
INSERT INTO address (address_id, city, street, zip_code, country_id) VALUES (30,
'Jeddah', 'MainStreet', '90597', 12);
```

```
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (1, 'New gaming laptops', '05/06/2014 09:00:00', '08/07/2021
10:27:00', 31);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (2, 'Back to school! New school materials', '09/07/2016
03:22:00', '06/16/2021 06:50:00', 32);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (3, 'New selection of UHD televisions', '04/16/2007 04:29:00',
'08/08/2021 01:19:00', 33);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (4, 'New iPhones now available', '03/16/2018 02:04:00',
'06/25/2021 03:17:00', 34);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (5, 'Discounts up to 50% on Xiaomi smarthphones', '02/08/2014
04:22:00', '07/28/2021 07:25:00', 35);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (6, 'Laptops on discount', '05/29/2003 03:21:00', '05/15/2021
09:56:00', 36);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (7, 'Discount on all school materials', '12/03/2016 09:06:00',
'04/26/2021 02:56:00', 37);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (8, 'New Lenovo laptops now available', '07/09/2006 06:18:00',
'06/28/2021 05:47:00', 38);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (9, 'All action movies on sale', '01/27/2000 05:34:00',
'04/14/2021 09:13:00', 39);
INSERT INTO advertisement (advertisement_id, title, begin_date, end_date,
photo_id) VALUES (10, 'Black Friday 2020', '12/05/2004 02:33:00', '04/26/2021
02:56:00', 40);
```

```
INSERT INTO apply_discount (item_id, discount_id) VALUES (10, 7);
INSERT INTO apply_discount (item_id, discount_id) VALUES (15, 10);
INSERT INTO apply_discount (item_id, discount_id) VALUES (19, 20);
INSERT INTO apply_discount (item_id, discount_id) VALUES (9, 5);
INSERT INTO apply_discount (item_id, discount_id) VALUES (16, 14);
INSERT INTO apply_discount (item_id, discount_id) VALUES (1, 3);
INSERT INTO apply_discount (item_id, discount_id) VALUES (11, 4);
INSERT INTO apply_discount (item_id, discount_id) VALUES (12, 7);
INSERT INTO apply_discount (item_id, discount_id) VALUES (4, 4);
INSERT INTO apply_discount (item_id, discount_id) VALUES (8, 9);
INSERT INTO apply_discount (item_id, discount_id) VALUES (10, 14);
INSERT INTO apply_discount (item_id, discount_id) VALUES (17, 2);
INSERT INTO apply_discount (item_id, discount_id) VALUES (20, 7);
INSERT INTO apply_discount (item_id, discount_id) VALUES (2, 15);
INSERT INTO apply_discount (item_id, discount_id) VALUES (3, 5);
INSERT INTO apply_discount (item_id, discount_id) VALUES (5, 11);
INSERT INTO apply_discount (item_id, discount_id) VALUES (14, 19);
INSERT INTO apply_discount (item_id, discount_id) VALUES (16, 8);
INSERT INTO apply_discount (item_id, discount_id) VALUES (20, 14);



INSERT INTO category_detail (category_id, detail_id) VALUES (1, 1);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 2);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 3);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 4);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 8);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 9);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 10);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 14);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (1, 21);
INSERT INTO category_detail (category_id, detail_id) VALUES (2, 5);
INSERT INTO category_detail (category_id, detail_id) VALUES (2, 15);
INSERT INTO category_detail (category_id, detail_id) VALUES (2, 16);
INSERT INTO category_detail (category_id, detail_id) VALUES (2, 22);
INSERT INTO category_detail (category_id, detail_id) VALUES (3, 17);
INSERT INTO category_detail (category_id, detail_id) VALUES (3, 18);
INSERT INTO category_detail (category_id, detail_id) VALUES (3, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (3, 20);
INSERT INTO category_detail (category_id, detail_id) VALUES (3, 21);
INSERT INTO category_detail (category_id, detail_id) VALUES (4, 5);
INSERT INTO category_detail (category_id, detail_id) VALUES (4, 6);
INSERT INTO category_detail (category_id, detail_id) VALUES (4, 7);
INSERT INTO category_detail (category_id, detail_id) VALUES (4, 22);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 1);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 2);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 3);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 8);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 9);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 10);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 11);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 12);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 13);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 14);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 18);
INSERT INTO category_detail (category_id, detail_id) VALUES (5, 19);
```

```
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 12);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 13);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 14);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 18);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 21);
INSERT INTO category_detail (category_id, detail_id) VALUES (6, 24);
INSERT INTO category_detail (category_id, detail_id) VALUES (7, 5);
INSERT INTO category_detail (category_id, detail_id) VALUES (7, 15);
INSERT INTO category_detail (category_id, detail_id) VALUES (7, 22);
INSERT INTO category_detail (category_id, detail_id) VALUES (7, 23);
INSERT INTO category_detail (category_id, detail_id) VALUES (8, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 5);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 17);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 18);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 23);
INSERT INTO category_detail (category_id, detail_id) VALUES (9, 24);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 9);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 10);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 17);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 18);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 19);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 20);
INSERT INTO category_detail (category_id, detail_id) VALUES (10, 21);
INSERT INTO category_detail (category_id, detail_id) VALUES (11, 5);
INSERT INTO category_detail (category_id, detail_id) VALUES (11, 6);
INSERT INTO category_detail (category_id, detail_id) VALUES (11, 22);
INSERT INTO category_detail (category_id, detail_id) VALUES (11, 23);



INSERT INTO item_photo (photo_id, item_id) VALUES (1, 1);
INSERT INTO item_photo (photo_id, item_id) VALUES (2, 1);
INSERT INTO item_photo (photo_id, item_id) VALUES (3, 2);
INSERT INTO item_photo (photo_id, item_id) VALUES (4, 2);
INSERT INTO item_photo (photo_id, item_id) VALUES (5, 3);
INSERT INTO item_photo (photo_id, item_id) VALUES (6, 4);
INSERT INTO item_photo (photo_id, item_id) VALUES (7, 5);
INSERT INTO item_photo (photo_id, item_id) VALUES (8, 6);
INSERT INTO item_photo (photo_id, item_id) VALUES (9, 6);
INSERT INTO item_photo (photo_id, item_id) VALUES (10, 7);
INSERT INTO item_photo (photo_id, item_id) VALUES (11, 7);
INSERT INTO item_photo (photo_id, item_id) VALUES (12, 7);
INSERT INTO item_photo (photo_id, item_id) VALUES (13, 8);
INSERT INTO item_photo (photo_id, item_id) VALUES (14, 9);
INSERT INTO item_photo (photo_id, item_id) VALUES (15, 10);
INSERT INTO item_photo (photo_id, item_id) VALUES (16, 11);
INSERT INTO item_photo (photo_id, item_id) VALUES (17, 11);
INSERT INTO item_photo (photo_id, item_id) VALUES (18, 12);
INSERT INTO item_photo (photo_id, item_id) VALUES (19, 13);
INSERT INTO item_photo (photo_id, item_id) VALUES (20, 14);
INSERT INTO item_photo (photo_id, item_id) VALUES (21, 15);
INSERT INTO item_photo (photo_id, item_id) VALUES (22, 15);
INSERT INTO item_photo (photo_id, item_id) VALUES (23, 16);
INSERT INTO item_photo (photo_id, item_id) VALUES (24, 16);
INSERT INTO item_photo (photo_id, item_id) VALUES (25, 17);
```

```
INSERT INTO item_photo (photo_id, item_id) VALUES (26, 18);
INSERT INTO item_photo (photo_id, item_id) VALUES (27, 19);
INSERT INTO item_photo (photo_id, item_id) VALUES (28, 20);
INSERT INTO item_photo (photo_id, item_id) VALUES (29, 20);
INSERT INTO item_photo (photo_id, item_id) VALUES (30, 20);



INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (1,
'Tom9', 'Freddy.Bergdahl@telefonica.com', 'Rachel', 'Hummel',
'E6MM7ndGb22eq3K5v1OACxLDFVmAKZmNgePm3AWrssEitMIOOwa72A1FVzt8FvymLylng2WG2cVTcnT
zVAlpsGLZiZeZdc4NiIv6mpE', False, True, '8128488.8', 41, 7, 7);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (2,
'Sophia', 'Mart.Hollman@msn.dk', 'Lia', 'Polti',
'TyNHIpa6MKX444MXmmO5qcNaC8QjHS7GLRQXjUKIlCQ2SSFGcKKcvCOppNSEAIygTFHOMMHiin4eznQ
2jYLdTSaUVIP6try8bSHprUuzfSjNnDVB1ZOpVqjkOLhEV6MnZS', False, True, '5223662.O',
42, 7, 7);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (3,
'Carlos83', 'gamingwithapotato@gmail.com', 'Herbert', 'Anderson',
'btWp5iKDSHBYQyZZ7sHIaOzDaATKZbeeSeCycslTiaXyeOemaY1vWYGWXHWijO2pKpmJex08LsomZ7y
SNzaMZbKIELfkYhIwoEuDJ2QhSQXY24EwJhcJ5ZQYYBQ7VVfTEXio2GcVABEQgeEoJJHfBb8gBpDtCT0
gmUhei1dmunu', False, False, '2938243.84', 43, 16, 16);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (4,
'Helma', 'FreddyWarner@gawab.ca', 'Fons', 'Lawton',
'LgMjfTOWQBTFcsYhiBNdXFIlPTHcC5As82bBHDLyKQedYTadPbM2bkUULpSdxr66wWfMMERFtypXB2h
LZNsfiftmhBqkBAXs268TL5ZIdpe7MhD8NanLtHMaVTDnSPcOT4Sf6aEbs3PlvaHL63jk6oUovkrdxHS
nHlkBMWAXTDTyh7kko5OyujOLmp', False, False, '9568377.54', 44, 21, 21);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (5,
'BessO', 'BiancaDulisse2@lycos.co.uk', 'Isaac', 'Cohen',
'FI5UQZrxYEDkEEQCz3E368V6jc3cwswNM3flBOOuWJveppbkE1MyQtYKQrsoZIdFMEb4YWmI3YypgPi
gOeOSJlVfUtOfb37ipo3phmOeFp8y2w2u637UCiD8781eoZISDTRC', False, False,
'7O89429.27', 45, NULL, 21);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (6,
'Suzanne36', 'RogierGuethlein4@weboffice.de', 'Javier', 'Petrzelka',
'FSBQAVmZ8SpmTRgXX1RRG8I2aHVo3g1SunXdfAqLYWpbGwB6OdGTbcUsDepaaOcaaW7s7df3cpMhoBe
RWv4TPcsOiromF1hmRj5Q1tgm8Ib8Eq2kpblgLVOJ8NphoTpSc3BMn4TeoR42', False, False,
'5O78O26.12', 46, 22, 22);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (7,
'Steph3', 'HToler@mymail.us', 'Mads', 'Storrs',
'pJ1ZJCvfbRXFfiXmgjulat3mZJWTlEXcNZAdYGXHlyCDvSkFfb3D81dy6xwWDdUIZSOLMdyKNE6VePj
pakevNbMbK4ZTL7BsdBJVOLvcA14imZ2D5cJxes8q8zjD4YGnde5w7OmYWjNiOcPuuoebcZAU4eZWU8n
DcggeZ3v2zll5Qee5BmxEOPczBvVuGtPG1TAA46NrxxTj14dN3', False, False, '1O8158O.6',
47, 22, 22);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (8,
'Piotr', 'Mandy.Fernandez5@freeweb.co.uk', 'Shermie', 'Helfrich',
'Y81DcKmovDAcVMYS47A6bsb1mSNqNlLvHXXrqNzvtxy4aK3uPZRwI5yVjstKp27NrRFZwdY5OTMKDh1
IOiwKCoEyvN81GjCvKVUtjYpeJUBLs1fgypQUSCj8ZiTvZ2zeCJngQUoWaZvkpO5WHl2Vn67gwP7ba03
3QApowfwnExc3dqdIoQIIk5XimAdcq1yt8lFRbFyHlvtQzRnNwVRSzoYnJQR8DcDSVqJLHwxOwNg6B7p
HqqXS', False, False, '2299613.26', 48, 22, 22);
```

```
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (9,
'Herb246', 'Trees.Gaskins5@gawab.es', 'Pieter', 'Orcutt',
'rNlwgEByc12R715pWnUlKzqtUI3kC3gvsFV2HPgRdUSi5dce64hZXZYXJ2zlWYjMLzPQFna6hEZ7QdO
CtPGUauUufDDOjrfIg8joPvALzxEaMjgyvnChJOXgJwpwKbx3KtNsy3UbgOL4jgDAV', False,
False, '2950352.32', 49, NULL, 22);
INSERT INTO users (user_id, username, email, first_name, last_name, password,
deleted, is_admin, balance, img, billing_address, shipping_address) VALUES (10,
NULL, NULL, NULL, NULL,
'xFUYINBeq4oPvnONs18WJZORqwjYYFPA4MuwQalgNjPODG2FEBcYjYN6azaYebnt6dfpg3AnBgfY4sq
4Erv8', True, False, NULL, NULL, NULL, NULL);



INSERT INTO wishlist (user_id, item_id, add_date) VALUES (7, 7, '04/20/2003
06:27:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (7, 11, '01/02/2006
06:58:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (4, 13, '01/16/2006
06:31:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (8, 3, '07/15/2009
08:22:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (9, 10, '02/28/2016
08:43:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (1, 20, '05/14/2019
06:21:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (5, 10, '12/05/2019
10:07:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (3, 19, '04/11/2013
02:53:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (10, 5, '09/10/2016
05:25:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (3, 6, '07/10/2010
00:23:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (7, 13, '03/16/2012
09:12:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (6, 17, '11/19/2019
03:25:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (7, 14, '09/15/2001
00:52:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (5, 4, '09/26/2019
02:12:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (5, 5, '03/03/2011
02:06:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (9, 13, '07/11/2003
08:52:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (8, 19, '02/07/2006
02:20:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (8, 1, '07/01/2003
05:38:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (5, 17, '02/04/2012
01:52:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (3, 2, '06/12/2005
06:06:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (3, 10, '11/25/2007
07:38:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (6, 13, '11/10/2005
06:16:00');
```

```
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (6, 14, '04/08/2018
03:34:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (1, 1, '04/23/2019
04:26:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (8, 5, '12/11/2005
05:06:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (9, 11, '04/28/2003
01:31:00');
INSERT INTO wishlist (user_id, item_id, add_date) VALUES (10, 20, '01/05/2017
07:53:00');



INSERT INTO ban (admin_id, user_id, date, reason) VALUES (1, 4, '07/12/2020
00:42:00', 'Offensive comments');
INSERT INTO ban (admin_id, user_id, date, reason) VALUES (2, 9, '10/04/2021
00:11:00', 'Racist comment');


INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 6,
'10/13/2010 09:58:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (3, 9,
'04/22/2000 01:43:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (2, 11,
'01/20/2002 08:38:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (6, 19,
'09/02/2016 00:17:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (10, 7,
'08/06/2006 01:44:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (5, 15,
'03/31/2011 00:32:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (2, 19,
'10/27/2002 01:18:00', 4);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (8, 1,
'10/15/2014 08:51:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (2, 8,
'12/28/2010 08:03:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (2, 12,
'02/17/2006 04:49:00', 10);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (2, 16,
'01/19/2012 03:22:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (1, 6,
'04/08/2001 00:08:00', 4);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (10, 11,
'06/30/2015 04:13:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (4, 2,
'10/08/2020 04:09:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (10, 8,
'02/19/2000 10:45:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (8, 11,
'01/02/2000 03:07:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 18,
'04/23/2002 02:32:00', 21);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 8,
'05/18/2008 04:32:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (8, 9,
'02/22/2013 08:25:00', 1);
```

```
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (8, 15,
'11/22/2017 06:07:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (6, 16,
'09/30/2017 07:47:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (4, 4,
'06/20/2006 08:52:00', 4);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (4, 14,
'11/15/2002 07:15:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (5, 19,
'10/13/2019 01:36:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (1, 2,
'03/11/2011 04:24:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 9,
'10/25/2004 05:52:00', 4);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 14,
'12/05/2011 02:34:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 15,
'11/18/2013 05:02:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (1, 8,
'06/26/2013 10:13:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (10, 17,
'04/02/2015 03:25:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 2,
'05/23/2019 10:22:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 4,
'08/31/2008 01:32:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 11,
'01/08/2009 00:30:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (6, 14,
'03/31/2005 07:53:00', 6);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (10, 1,
'08/17/2001 02:22:00', 5);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (5, 10,
'09/28/2014 10:35:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (3, 19,
'09/24/2019 02:22:00', 6);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (8, 5,
'03/26/2020 07:47:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (6, 3,
'06/04/2003 08:48:00', 3);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (7, 5,
'05/20/2007 01:16:00', 2);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 7,
'06/06/2012 01:53:00', 1);
INSERT INTO cart (user_id, item_id, add_date, quantity) VALUES (9, 13,
'05/09/2010 09:42:00', 3);



INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (2, 8, 1, 4, 'Discount',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (2, NULL, 2, 13, 'Stock',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (2, NULL, 3, 13, 'Stock',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (8, NULL, 4, 13, 'Stock',False);
```

```
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (8, NULL, 5, 3, 'Stock',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (8, 1, 6, 9, 'Discount',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (6, NULL, 7, 9, 'Stock',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (6, 1, 8, 9, 'Discount',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (6, NULL, 9, 11, 'Stock',False);
INSERT INTO notification (user_id, discount_id, notification_id, item_id, type,
is_seen) VALUES (8, 1, 10, 19, 'Discount',False);



INSERT INTO purchase (purchase_id, user_id, date) VALUES (1, 3, '12/27/2003
09:09:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (2, 3, '07/29/2012
02:09:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (3, 4, '05/30/2017
01:17:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (4, 5, '09/12/2000
07:13:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (5, 6, '08/12/2006
00:14:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (6, 6, '02/10/2011
04:52:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (7, 6, '07/01/2000
04:25:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (8, 8, '05/28/2017
01:44:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (9, 9, '08/25/2003
04:53:00');
INSERT INTO purchase (purchase_id, user_id, date) VALUES (10, 10, '10/10/2015
05:38:00');



INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (8, 11,
05.37, 1);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (5, 1,
4174.27, 2);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (6, 7,
65671, 1);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (1, 9,
78777.62, 3);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (7, 19,
37081.80, 1);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (5, 2,
3791.88, 2);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (2, 11,
21.62, 5);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (3, 18,
4.05, 3);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (1, 2,
1.35, 2);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (5, 7,
562.90, 1);
```

```
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (7, 9, 25469.87, 2);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (2, 19, 59835.15, 1);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (4, 3, 47.88, 2);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (1, 12, 0.69, 4);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (3, 13, 33242.28, 3);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (1, 20, 1962.65, 6);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (8, 4, 375.97, 3);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (3, 5, 59.42, 6);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (5, 12, 54101.88, 10);
INSERT INTO purchase_item (purchase_id, item_id, price, quantity) VALUES (9, 17, 5875.08, 1);



INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (1, 3, 1, 'Very good!', '08/09/2007 09:48:00', 4);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (2, 4, 1, 'Expensive but worth every penny', '08/30/2007 10:00:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (3, 3, 2, 'Awful! Do not buy this', '03/17/2000 07:05:00', 1);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (4, 5, 2, 'This laptop is perfect for what I want. I can take it with me
and work anywhere! Nice battery duration also.', '08/24/2006 02:38:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (9, 4, 2, 'A must buy for whoever wants a small and light laptop that can
do the work', '07/30/2010 05:21:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (5, 4, 3, 'I fell in love with this book. Totally recommend it',
'09/06/2011 06:22:00', 4);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (6, 5, 4, 'I feel like a proper business man after reading this',
'02/24/2009 00:45:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (7, 3, 5, 'It broke after a day of use. Do not buy this, material is too
cheap', '04/25/2000 09:30:00', 1);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (8, 7, 6, 'Had to return this. Does not work well, at all', '06/28/2018
10:33:00', 1);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (10, 8, 9, 'its a good phone, better than most but wayyy too expensive',
'03/03/2006 07:52:00', 3);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (11, 3, 9, 'Ive always bought iPhones for years and this one doesnt
disappoint! It is a bit expensive but you cannot find anything better out
there', '03/26/2003 04:18:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (12, 6, 9, 'apple what are you doing???', '12/17/2001 00:38:00', 1);
```

```
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (13, 8, 10, 'I was enjoying it, until the folding part of the screen
started getting weird', '05/29/2013 10:34:00', 2);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (14, 4, 11, 'perfect weight and sound. always use them to listen to music
while im studying', '12/11/2005 00:43:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (15, 2, 16, 'Ive been gaming on this tv and its just a blast. Amazing
quality', '01/22/2016 01:17:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (16, 6, 17, 'the image quality is just stunning. best tv ive ever
bought', '04/30/2013 09:43:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (17, 8, 17, 'This is literally the future', '04/21/2000 03:13:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (18, 6, 18, 'Not as good as the other ones, but still a fun ride',
'05/11/2002 01:41:00', 3);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (19, 6, 19, 'EPIIIICCCCC!! MUST WATCH', '05/11/2006 04:41:00', 5);
INSERT INTO review (review_id, user_id, item_id, comment_text, date, rating)
VALUES (20, 7, 20, 'Average movie, not much to say', '09/06/2013 06:28:00', 3);
```

Revision history

Changes made to the first submission:

1. Item 1
2. ..

GROUP2111, 18/04/2021

- Diogo Guimarães do Rosário, up201806582@fe.up.pt (Editor)
- Henrique Melo Ribeiro, up201806529@fe.up.pt
- Davide António Ferreira Castro, up201806512@fe.up.pt
- João Alexandre Lobo Cardoso, up201806531@fe.up.pt