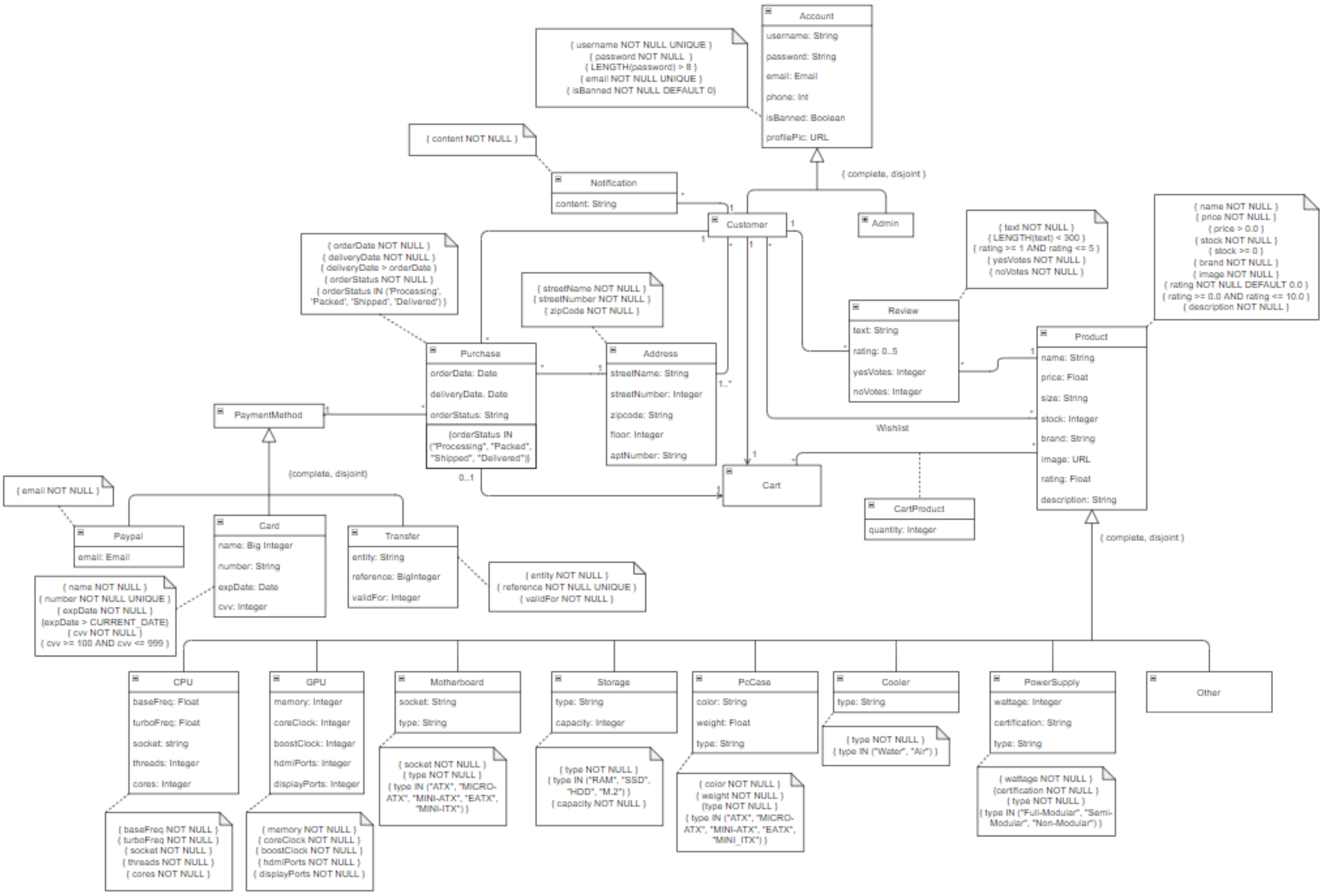
EBD: Database Specification Component

A4: Conceptual Data Model

The Conceptual Data Model represents all relations created on a database and explicitly show how they are connected, through an UML diagram.

1. Class diagram



2. Additional Business Rules

> No additional rules were created. All business rules are placed on the UML.

A5: Relational Schema, validation and schema refinement

This artifact contains the Relation Model mapping of the UML previously displayed on the A4, alongside the new created domains, to define new data types. Every single relation is parsed (Functional Dependencies are specified) and validated in order to enforce normalization.

1. Relational Schema

Relation reference	Relation Compact Notation

Relation reference	Relation Compact Notation
R01	Account(id, username **NN UK**, password **NN CK** LENGTH(password) > 8, email **NN UK**, phone, isBanned **NN DF** 0, profilePic **DF** “images/default.png”)
R02	Customer(id → Account, id_Cart → Cart **NN UK**)
R03	Admin(id → Account)
R04	Product(id, name **NN UK**, price **NN CK** price > 0.0, size, stock **NN CK** stock >= 0, brand **NN**, image **NN**, rating **NN DF** 0.0 **CK** rating >= 0.0 AND rating <= 5.0, description **NN**)
R05	CPU(id → Product, baseFreq **NN**, turboFreq **NN**, socket **NN**, threads **NN**, cores **NN**)
R06	GPU(id → Product, memory **NN**, coreClock **NN**, boostClock **NN**, hdmiPorts **NN**, displayPorts **NN**)
R07	Motherboard(id → Product, socket **NN**, type **NN CK** type IN MotherboardType)
R08	Storage(id → Product, capacity **NN**, type **NN CK** type IN StorageType)
R09	PcCase(id → Product, color **NN**, weight **NN**, type **NN CK** type IN MotherboardType)
R10	Cooler(id → Product, type **NN CK** type IN CoolerType)
R11	PowerSupply(id → Product, wattage **NN**, certification **NN**, type **NN CK** type IN PowerSupplyType)
R12	Other(id → Product)
R13	Review(id, text **NN CK** LENGTH(text) < 300, rating **NN CK** rating >= 1 AND rating <= 5, yesVotes **NN DF** 0, noVotes **NN DF** 0, id_Customer → Customer **NN**, id_Product → Product **NN**)
R14	Cart(id)
R15	Address(id, streetName **NN**, streetNumber **NN**, zipcode **NN**, floor, aptNumber)
R16	Purchase(id, orderDate **NN**, deliveryDate **NN CK** deliveryDate > orderDate, orderStatus **NN CK** orderStatus IN OrderStatusType, id_Customer → Customer **NN**, id_Address → Address **NN**, id_PaymentMethod → PaymentMethod **NN**, id_Cart → Cart **NN UK**)
R17	PaymentMethod(id)
R18	Paypal(id → PaymentMethod, email **NN UK**)
R19	Card(id → PaymentMethod, name **NN**, number **NN UK**, expDate **NN CK** expDate > CURRENT_DATE, cvv **NN CK** cvv >= 100 AND cvv <= 999)
R20	Transfer(id → PaymentMethod, entity **NN**, reference **NN UK** validFor **NN DF** 24)
R21	Notification(id, content **NN**, id_Customer → Customer)
R22	CustomerAddress(id_Customer → Customer, id_Address → Address)
R23	Wishlist(id_Customer → Customer, id_Product → Product)

Relation reference	Relation Compact Notation
R24	CartProduct(id_Cart → Cart, id_Product → Product, quantity **NN**)

> Note: **UK** = **UNIQUE**, **NN** = **NOT NULL**, **DF** = **DEFAULT**, **CK** = **CHECK**

1.1 Generalizations Justifications

Generalization Relation	Specializations	Properties	Justification
R01	R02, R03	{complete, disjoint}	Every account created(complete) is either a customer or an admin account(disjoint).
R04	R05, R06, R07, R08, R09, R10, R11, R12	{complete, disjoint}	Every product(complete)belongs to a single category at a time(disjoint).
R17	R18, R19, R20	{complete, disjoint}	Every payment method has to be one of the following three(Paypal, Card or Transfer). No payment method can be in two specializations at the same time.

2. Domains

Domain Name	Domain Specification
MotherboardType	ENUM('ATX', 'MICRO-ATX', 'MINI-ATX', 'EATX', 'MINI-ITX')
StorageType	ENUM('RAM', 'SSD', 'HDD', 'M.2')
CoolerType	ENUM('Water', 'Air')
PowerSupplyType	ENUM('Full-Modular', 'Semi-Modular', 'Non-modular')
OrderStatusType	ENUM('Processing', 'Packed', Shipped', 'Delivered')

3. Schema validation

Table R01	**Account**
Keys	{id}, {username}, {email}
Functional Dependencies:	
FD0101	{id} → {username, password, email, phone, isBanned, profilePic}
FD0102	{username} → {id, password, email, phone, isBanned, profilePic}
FD0103	{email} → {id, username, password, phone, isBanned, profilePic}
Normal Form	BCNF

Table R02	**Customer**

Table R02	**Customer**
Keys	{id}, {id_Cart}
Functional Dependencies:	
FD0201	{id} → {id_Cart}
FD0201	{id_Cart} → {id}
Normal Form	BCNF

Table R03	**Admin**
Keys	{id}
Functional Dependencies:	None
Normal Form	BCNF

Table R04	**Product**
Keys	{id}, {name}
Functional Dependencies:	
FD0401	{id} → {name, price, size, stock, brand, image, rating, description}
FD0402	{name} → {id, price, size, stock, brand, image, rating, description}
Normal Form	BCNF

Table R05	**CPU**
Keys	{id}
Functional Dependencies:	
FD0501	{id} → {baseFreq, turboFreq, socket, threads, cores}
Normal Form	BCNF

Table R06	**GPU**
Keys	{id}
Functional Dependencies:	
FD0601	{id} → {memory, coreClock, boostClock, hdmiPorts, displayPorts}
Normal Form	BCNF

Table R07	**Motherboard**
Keys	{id}
Functional Dependencies:	
FD0701	{id} → {socket, type}
Normal Form	BCNF

Table R08	**Storage**
Keys	{id}
Functional Dependencies:	
FD0801	{id} → {size, type}
Normal Form	BCNF

Table R09	**PcCase**
Keys	{id}
Functional Dependencies:	
FD0901	{id} → {color, weight, type}
Normal Form	BCNF

Table R10	**Cooler**
Keys	{id}
Functional Dependencies:	
FD1001	{id} → {type}
Normal Form	BCNF

Table R11	**PowerSupply**
Keys	{id}
Functional Dependencies:	
FD1101	{id} → {wattage, certification, type}
Normal Form	BCNF

Table R12	**Other**
Keys	{id}
Functional Dependencies:	None
Normal Form	BCNF

Table R13	**Review**
Keys	{id}
Functional Dependencies:	
FD1301	{id} → {text, rating, yesVotes, noVotes, id_Customer, id_Product}
Normal Form	BCNF

Table R14	**Cart**
Keys	{id}

Table R14	**Cart**
Functional Dependencies:	None
Normal Form	BCNF

Table R15	**Address**
Keys	{id}
Functional Dependencies:	
FD1501	{id} → {streetName, streetNumber, zipcode, floor, aptNumber}
Normal Form	BCNF

Table R16	**Purchase**
Keys	{id}, {id_Cart}
Functional Dependencies:	
FD1601	{id} → {orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart}
FD1602	{id_Cart} → {id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod}
Normal Form	BCNF

Table R17	**PaymentMethod**
Keys	{id}
Functional Dependencies:	None
Normal Form	BCNF

Table R18	**Paypal**
Keys	{id}, {email}
Functional Dependencies:	
FD1801	{id} → {email}
FD1802	{email} → {id}
Normal Form	BCNF

Table R19	**Card**
Keys	{id}, {number}
Functional Dependencies:	
FD1901	{id} → {name, number, expDate, cvv}
FD1902	{number} → {id, name, expDate, cvv}

Table R19	**Card**
Normal Form	BCNF

Table R20	**Transfer**
Keys	{id}, {reference}
Functional Dependencies:	
FD2001	{id} → {entity, reference, validFor}
FD2002	{reference} → {id, entity, validFor}
Normal Form	BCNF

Table R21	**Notification**
Keys	{id}
Functional Dependencies:	
FD2101	{id} → {content, id_Customer}
Normal Form	BCNF

Table R22	**CustomerAddress**
Keys	{id_Customer, id_Address}
Functional Dependencies:	None
Normal Form	BCNF

Table R23	**Wishlist**
Keys	{id_Customer, id_Product}
Functional Dependencies:	None
Normal Form	BCNF

Table R24	**CartProduct**
Keys	{id_Cart, id_Product}
Functional Dependencies:	
FD2401	{id_Cart, id_Product} → {quantity}
Normal Form	BCNF

Because the schema was built on relations that follow BCNF, the schema itself is in BCNF as well, which means there is no need for further normalization.

A6: Indexes, triggers, transactions and database population

This artifact contains the expected size and growth over time of each relation created on the database, proposed performance indexes to, improve database access time and full-text search indexes as well. The SQL schema creation and population script are both displayed in the end.

1. Database Workload

Relation reference	Relation Name	Order of magnitude	Estimated growth
R01	Account	tens of thousands	hundreds per month
R02	Customer	tens of thousands	hundreds per month
R03	Admin	tens	units per year
R04	Product	hundreds	dozens per month
R05	CPU	tens	units per month
R06	GPU	tens	units per month
R07	Motherboard	tens	units per month
R08	Storage	tens	units per month
R09	PcCase	tens	units per month
R10	Cooler	tens	units per month
R11	PowerSupply	tens	units per month
R12	Other	tens	units per month
R13	Review	thousands	hundreds per month
R14	Cart	tens of thousands	hundreds per month
R15	Address	thousands	hundreds per month
R16	Purchase	thousands	dozens per day
R17	PaymentMethod	thousands	dozens per day
R18	Paypal	thousands	dozens per day
R19	Card	thousands	dozens per day
R20	Transfer	thousands	dozens per day
R21	Notification	tens of thousands	hundreds per month
R22	CustomerAddress	thousands	hundreds per month
R23	Wishlist	tens of thousands	hundreds per day
R24	CartProduct	thousands	hundreds per day

2. Proposed Indices

2.1. Performance Indices

Index	**IDX01**
Relation	Purchase
Attribute	id_Customer
Type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	The Purchase table will feature every order and thus will be quite large. Several queries might require filtering order made by a specific user. Filtering is done by exact match, thus an hash type index would be best suited. However, since we also want to apply clustering based on this index, and clustering is not possible on hash type indexes, we opted for a b-tree index. With medium cardinality it's a good candidate for clustering.
SQL	`CREATE INDEX order_user ON Purchase(id_Customer); CLUSTER Purchase USING order_user;`

Index	**IDX02**
Relation	Product
Attribute	price
Type	B-tree
Cardinality	high
Clustering	No
Justification	B-Tree indexing will helps us sort and filter products by price.
SQL	`CREATE INDEX product_price ON Product(price);`

Index	**IDX03**
Relation	Product
Attribute	brand
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering products by brand. Filtering is exact match, hence Hash.
SQL	`CREATE INDEX product_brand ON Product USING hash (brand);`

Index	**IDX04**
Relation	Cooler
Attribute	type

Index	**IDX04**
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering Cooler by type of cooling, Filtering is exact match so we use Hash
SQL	`CREATE INDEX cooler_type ON Cooler USING hash(type);`

Index	**IDX05**
Relation	PowerSupply
Attribute	type
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering PowerSupply by type, Filtering is exact match so we use Hash
SQL	`CREATE INDEX supply_type ON PowerSupply USING hash(type);`

Index	**IDX06**
Relation	PcCase
Attribute	type
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering PcCase by type, Filtering is exact match so we use Hash
SQL	`CREATE INDEX case_type ON PcCase USING hash(type);`

Index	**IDX07**
Relation	Storage
Attribute	type
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering Storage by type, Filtering is exact match so we use Hash
SQL	`CREATE INDEX storage_type ON Storage USING hash(type);`

Index	**IDX08**

Index	**IDX08**
Relation	Motherboard
Attribute	type
Type	Hash
Cardinality	Low
Clustering	No
Justification	Used in filtering Motherboard by type, Filtering is exact match so we use Hash
SQL	`CREATE INDEX mb_type ON Motherboard USING hash(type);`

Index	**IDX09**
Relation	Review
Attribute	id_Customer
Type	Hash
Cardinality	Medium
Clustering	No
Justification	To quickly filter reviews by customer
SQL	`CREATE INDEX review_author ON Review USING hash(id_Customer);`

Index	**IDX10**
Relation	Review
Attribute	id_Product
Type	Hash
Cardinality	Medium
Clustering	Yes
Justification	Filtering and listening reviews by product. Since we might want to frequently list reviews for a specific product, and update frequency should be low it makes sense to cluster.
SQL	`CREATE INDEX review_product ON Review (id_Product); CLUSTER Order USING order_user;`

Index	**IDX11**
Relation	Review
Attribute	rating
Type	B-Tree
Cardinality	Low

Index	**IDX11**
Clustering	No
Justification	To quickly filter and order reviews by customer
SQL	`CREATE INDEX review_rating ON Review(rating);`

2.2. Full-text Search Indices

Index	**IDX01**
Relation	Product
Attribute	Name, Brand
Type	GIN
Clustering	No
Justification	Should provide full-text search features to look for products based on name or brand. The index type is GIN because the indexed fields are not expected to change often.
SQL	`CREATE INDEX search_idx ON Product USING GIN (tsvectors); ALTER TABLE Product ADD COLUMN tsvectors TSVECTOR; CREATE FUNCTION productSearchUpdate() RETURNS TRIGGER AS $$ BEGIN IF TG_OP = 'INSERT' THEN NEW.tsvectors = (setweight(to_tsvector('english', NEW.name), 'A') \|\| setweight(to_tsvector('english', NEW.brand), 'B')); END IF; IF TG_OP = 'UPDATE' THEN IF (NEW.name <> OLD.name OR NEW.brand <> OLD.brand) THEN NEW.tsvectors = (setweight(to_tsvector('english', NEW.name), 'A') \|\| setweight(to_tsvector('english', NEW.brand), 'B')); END IF; END IF; RETURN NEW; END $$ LANGUAGE plpgsql; CREATE TRIGGER productSearchUpdate BEFORE INSERT OR UPDATE ON Product FOR EACH ROW EXECUTE PROCEDURE productSearchUpdate(); CREATE INDEX search_idx ON work USING GIN (tsvectors);`

3. Triggers

Trigger	**TRIGGER01**
Description	Updates the rating of a product when a review is added or changed
SQL	`CREATE FUNCTION updateProductRating() RETURNS TRIGGER AS $BODY$ BEGIN Update Product Set Product.rating = (SELECT avg(review.rating) from Review where id_Product = NEW.id_Product) where Product.id = NEW.id_Product; RETURN NULL; END $BODY$ LANGUAGE plpgsql; CREATE TRIGGER updateProductRating AFTER INSERT OR UPDATE ON Review FOR EACH ROW EXECUTE PROCEDURE updateProductRating();`

Trigger	**TRIGGER02**
Description	Prevents a user from adding more items to the cart than the available stock

Trigger	TRIGGER02
SQL	`CREATE FUNCTION verificaStock() RETURNS TRIGGER AS $BODY$ BEGIN IF NOT EXISTS(SELECT * from Product WHERE NEW.id_Product = Product.id_Product AND NEW.quantity <= Product.quantity) THEN RAISE EXCEPTION 'You can NOT add that much quantity to the cart'; END IF; RETURN NULL; END $BODY$ LANGUAGE plpgsql; CREATE TRIGGER verificaStock BEFORE INSERT ON CartProduct FOR EACH ROW EXECUTE PROCEDURE verificaStock();`

Trigger	TRIGGER03
Description	Prevents a banned user from writting a review
SQL	`CREATE FUNCTION blockBannedUsers() RETURN TRIGGER AS $BODY$ BEGIN IF NOT EXISTS(SELECT isBanned from Account WHERE Account.id = New.id_Customer AND isBanned = false) THEN RAISE EXCEPTION 'You can NOT write reviews as you´ve been banned from the website'; END IF; RETURN NULL; END $BODY$ LANGUAGE plpgsql; CREATE TRIGGER blockBannedUsers BEFORE INSERT ON Review FOR EACH ROW EXECUTE PROCEDURE blockBannedUsers();`

Trigger	TRIGGER04
Description	Sends a notification to the customer when there is a change of status in their order
SQL	`CREATE FUNCTION orderStatusNotification() RETURNS TRIGGER AS $BODY$ BEGIN IF EXISTS (SELECT * from Customer where New.id_Customer = Customer.id) THEN INSERT INTO Notification(content, id_Customer) VALUES ("Your order status has been updated" , New.id_Customer); END IF;RETURN NULL; END $BODY$ LANGUAGE plpgsql; CREATE TRIGGER orderStatusNotification AFTER UPDATE ON Purchase FOR EACH ROW EXECUTE PROCEDURE orderStatusNotification();`

4. Transactions

Transaction	TRAN01
Description	Get current cart products
Justification	In the middle of a transaction, information about some products in a cart may be added or removed, so that information in both selects may be different, which results in a Phantom Read. It's READ ONLY because only uses selects query.
SQL	`BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY; SELECT Cart.id, Product.name, CartProduct.quantity FROM CartProduct INNER JOIN Product ON Product.id = CartProduct.id_Product INNER JOIN Cart ON Cart.id = CartProduct.id_Cart; END TRANSACTION;`

Annex A. SQL Code

A.1. Database schema

```
drop table if exists Account CASCADE;
drop table if exists Address CASCADE;
drop table if exists Admin CASCADE;
drop table if exists Card CASCADE;
drop table if exists Cart CASCADE;
drop table if exists CartProduct CASCADE;
drop table if exists Cooler CASCADE;
drop table if exists CPU CASCADE;
drop table if exists Customer CASCADE;
drop table if exists CustomerAddress CASCADE;
drop table if exists GPU CASCADE;
drop table if exists Motherboard CASCADE;
drop table if exists Notification CASCADE;
drop table if exists Other CASCADE;
drop table if exists PaymentMethod CASCADE;
drop table if exists Paypal CASCADE;
drop table if exists PcCase CASCADE;
drop table if exists PowerSupply CASCADE;
drop table if exists Product CASCADE;
drop table if exists Purchase CASCADE;
drop table if exists Review CASCADE;
drop table if exists Storage CASCADE;
drop table if exists Transfer CASCADE;
drop table if exists Wishlist CASCADE;
drop trigger if exists updateProductRating on Review;
drop trigger if exists verificaStock on CartProduct;
drop trigger if exists blockBannedUsers on Review;
drop trigger if exists orderStatusNotification on Purchase;
drop trigger if exists productSearchUpdate on Product;
drop function if exists updateProductRating;
drop function if exists verificaStock;
drop function if exists blockBannedUsers;
drop function if exists orderStatusNotification;
drop function if exists productSearchUpdate;
drop index if exists acc_id;
drop index if exists product_price;
drop index if exists product_brand;
drop index if exists cooler_type;
drop index if exists powersupply_type;
drop index if exists case_type;
drop index if exists storage_type;
drop index if exists mb_type;
drop index if exists review_author;
drop index if exists review_product;
drop index if exists review_rating;
drop index if exists order_user;
drop index if exists search_idx;
drop type if exists MotherboardType;
drop type if exists StorageType;
drop type if exists CoolerType;
drop type if exists PowerSupplyType;
drop type if exists OrderStatusType;
```

```
CREATE TYPE MotherboardType as ENUM('ATX', 'MICRO-ATX', 'MINI-ATX', 'EATX', 'MINI-ITX');
CREATE TYPE StorageType as ENUM('RAM', 'SSD', 'HDD', 'M.2');
CREATE TYPE CoolerType as ENUM('Water', 'Air');
CREATE TYPE PowerSupplyType as ENUM('Full-Modular', 'Semi-Modular', 'Non-Modular');
CREATE TYPE OrderStatusType as ENUM ('Processing', 'Packed', 'Shipped', 'Delivered');

CREATE TABLE Account(
    id SERIAL,
    username TEXT NOT NULL UNIQUE,
    password TEXT NOT NULL check (LENGTH(password) > 8),
    email TEXT NOT NULL UNIQUE,
    phone INTEGER,
    profilePic TEXT DEFAULT 'images/default.jpg',
    isBanned BOOLEAN DEFAULT false,

    CONSTRAINT Account_PK PRIMARY KEY (id)
);

CREATE TABLE Cart(
    id SERIAL,

    CONSTRAINT Cart_PK PRIMARY KEY(id)
);

CREATE TABLE Customer(
    id INTEGER,
    id_Cart INTEGER NOT NULL UNIQUE,

    CONSTRAINT Customer_PK PRIMARY KEY(id),
    CONSTRAINT Customer_FK1 FOREIGN KEY(id) REFERENCES Account(id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT Customer_FK2 FOREIGN KEY(id_Cart) REFERENCES Cart(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Admin(
    id INTEGER,

    CONSTRAINT Admin_PK PRIMARY KEY(id),
    CONSTRAINT Admin_FK FOREIGN KEY(id) REFERENCES Account(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Product(
    id SERIAL,
    name TEXT NOT NULL UNIQUE,
    price FLOAT NOT NULL CHECK(price > 0.0),
    size TEXT,
    stock INTEGER NOT NULL CHECK(stock >= 0),
    brand TEXT NOT NULL,
    image TEXT NOT NULL,
```

```sql
    rating FLOAT NOT NULL DEFAULT 0.0 CHECK(rating >= 0.0 AND rating <= 5.0),
    description TEXT NOT NULL,

    CONSTRAINT Product_PK PRIMARY KEY(id)
);

CREATE TABLE CPU(
    id INTEGER,
    baseFreq FLOAT NOT NULL,
    turboFreq FLOAT NOT NULL,
    socket TEXT NOT NULL,
    threads INTEGER NOT NULL,
    cores INTEGER NOT NULL,

    CONSTRAINT CPU_PK PRIMARY KEY(id),
    CONSTRAINT CPU_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE GPU(
    id INTEGER,
    memory INTEGER NOT NULL,
    coreClock INTEGER NOT NULL,
    boostClock INTEGER NOT NULL,
    hdmiPorts INTEGER NOT NULL,
    displayPorts INTEGER NOT NULL,

    CONSTRAINT GPU_PK PRIMARY KEY(id),
    CONSTRAINT GPU_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Motherboard(
    id INTEGER,
    socket TEXT NOT NULL,
    type MotherboardType NOT NULL,

    CONSTRAINT Motherboard_PK PRIMARY KEY(id),
    CONSTRAINT Motherboard_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Storage(
    id INTEGER,
    capacity INTEGER NOT NULL,
    type StorageType NOT NULL,

    CONSTRAINT Storage_PK PRIMARY KEY(id),
    CONSTRAINT Storage_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PcCase(
    id INTEGER,
    color TEXT NOT NULL,
    weight TEXT NOT NULL,
    type MotherboardType NOT NULL,

    CONSTRAINT PcCase_PK PRIMARY KEY(id),
    CONSTRAINT PcCase_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Cooler(
    id INTEGER,
    type CoolerType NOT NULL,

    CONSTRAINT Cooler_PK PRIMARY KEY(id),
    CONSTRAINT Cooler_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PowerSupply(
    id INTEGER,
    wattage INTEGER NOT NULL,
    type PowerSupplyType NOT NULL,
    certification TEXT NOT NULL,

    CONSTRAINT Powersupply_PK PRIMARY KEY(id),
    CONSTRAINT Powersupply_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Other(
    id INTEGER,

    CONSTRAINT Other_PK PRIMARY KEY(id),
    CONSTRAINT Other_FK FOREIGN KEY(id) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Review(
    id SERIAL,
    text TEXT NOT NULL CHECK (LENGTH(text) < 300),
    rating INTEGER NOT NULL CHECK (rating >= 1 AND rating <= 5),
    yesVotes INTEGER NOT NULL DEFAULT 0,
    noVotes INTEGER NOT NULL DEFAULT 0,
    id_Customer INTEGER NOT NULL,
    id_Product INTEGER NOT NULL,

    CONSTRAINT Review_PK PRIMARY KEY(id),
    CONSTRAINT Review_FK1 FOREIGN KEY(id_Customer) REFERENCES Customer(id) ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT Review_FK2 FOREIGN KEY(id_Product) REFERENCES Product(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Address(
    id SERIAL,
    streetName TEXT NOT NULL,
    streetNumber INTEGER NOT NULL,
    zipcode TEXT NOT NULL,
    floor INTEGER,
    aptNumber TEXT,

    CONSTRAINT Address_PK PRIMARY KEY(id)
);

CREATE TABLE PaymentMethod(
    id SERIAL NOT NULL,
    CONSTRAINT PaymentMethod_PK PRIMARY KEY(id)
);

CREATE TABLE Purchase(
    id SERIAL,
    orderDate DATE NOT NULL,
    deliveryDate DATE NOT NULL,
    orderStatus OrderStatusType NOT NULL,
    id_Customer INTEGER NOT NULL,
    id_Address INTEGER NOT NULL,
    id_PaymentMethod INTEGER NOT NULL,
    id_Cart INTEGER NOT NULL UNIQUE,

    CHECK (deliveryDate > orderDate),

    CONSTRAINT Order_PK PRIMARY KEY(id),
    CONSTRAINT Order_FK1 FOREIGN KEY(id_Customer) REFERENCES Customer(id) ON
DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT Order_FK2 FOREIGN KEY(id_Address) REFERENCES Address(id) ON DELETE
SET NULL ON UPDATE CASCADE,
    CONSTRAINT Order_FK3 FOREIGN KEY(id_PaymentMethod) REFERENCES
PaymentMethod(id) ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT Order_FK4 FOREIGN KEY(id_Cart) REFERENCES Cart(id) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Paypal(
    id INTEGER,
    email TEXT NOT NULL UNIQUE,

    CONSTRAINT Paypal_PK PRIMARY KEY(id),
    CONSTRAINT Paypal_FK FOREIGN KEY(id) REFERENCES PaymentMethod(id) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Card(
    id INTEGER,
    name TEXT NOT NULL,
    number BIGINT NOT NULL UNIQUE,
    expDate DATE NOT NULL CHECK (expDate > CURRENT_DATE),
    cvv INTEGER NOT NULL CHECK (cvv >= 100 AND cvv <= 999),
    CONSTRAINT Card_PK PRIMARY KEY(id),
    CONSTRAINT Card_FK FOREIGN KEY(id) REFERENCES PaymentMethod(id) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Transfer(
    id INTEGER,
    entity INTEGER NOT NULL,
    reference BIGINT NOT NULL UNIQUE,
    validFor INTEGER NOT NULL DEFAULT 24,

    CONSTRAINT Transfer_PK PRIMARY KEY(id),
    CONSTRAINT Transfer_FK FOREIGN KEY(id) REFERENCES PaymentMethod(id) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Notification(
    id SERIAL,
    content TEXT NOT NULL,
    id_Customer INTEGER NOT NULL,

    CONSTRAINT Notification_PK PRIMARY KEY(id),
    CONSTRAINT Notification_FK FOREIGN KEY(id_Customer) REFERENCES Customer(id) ON
DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE CustomerAddress(
    id_Customer INTEGER,
    id_Address INTEGER,

    CONSTRAINT CustomerAddress_PK PRIMARY KEY (id_Customer, id_Address),
    CONSTRAINT CustomerAddress_FK1 FOREIGN KEY(id_Customer) REFERENCES
Customer(id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT CustomerAddress_FK2 FOREIGN KEY(id_Address) REFERENCES Address(id)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Wishlist(
    id_Customer INTEGER,
    id_Product INTEGER,

    CONSTRAINT Wishlist_PK PRIMARY KEY (id_Customer, id_Product),
    CONSTRAINT Wishlist_FK1 FOREIGN KEY(id_Customer) REFERENCES Customer(id) ON
DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT Wishlist_FK2 FOREIGN KEY(id_Product) REFERENCES Product(id) ON
DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE CartProduct(
    id_Cart INTEGER,
    id_Product INTEGER,
    quantity INTEGER NOT NULL,
    CONSTRAINT CartProduct_PK PRIMARY KEY (id_Cart, id_Product),
    CONSTRAINT CartProduct_FK1 FOREIGN KEY(id_Cart) REFERENCES Cart(id) ON DELETE
CASCADE ON UPDATE CASCADE,
    CONSTRAINT CartProduct_FK2 FOREIGN KEY(id_Product) REFERENCES Product(id) ON
DELETE CASCADE ON UPDATE CASCADE
);

--------------------------------------------------------
-- INDEXES
--------------------------------------------------------

CREATE INDEX order_user ON Purchase(id_Customer);
CLUSTER Purchase USING order_user;

CREATE INDEX product_price ON Product(price);

CREATE INDEX product_brand ON Product USING hash (brand);

CREATE INDEX cooler_type ON Cooler USING hash(type);

CREATE INDEX supply_type ON PowerSupply USING hash(type);

CREATE INDEX case_type ON PcCase USING hash(type);

CREATE INDEX storage_type ON Storage USING hash(type);

CREATE INDEX mb_type ON Motherboard USING hash(type);

CREATE INDEX review_author ON Review USING hash(id_Customer);

CREATE INDEX review_product ON Review (id_Product);
CLUSTER Purchase USING order_user;

CREATE INDEX review_rating ON Review(rating);

ALTER TABLE Product ADD COLUMN tsvectors TSVECTOR;

CREATE FUNCTION productSearchUpdate() RETURNS TRIGGER AS $$
BEGIN
    IF TG_OP = 'INSERT' THEN
        NEW.tsvectors = (
         setweight(to_tsvector('english', NEW.name), 'A') ||
         setweight(to_tsvector('english', NEW.brand), 'B')
        );
    END IF;
    IF TG_OP = 'UPDATE' THEN
         IF (NEW.name <> OLD.name OR NEW.brand <> OLD.brand) THEN
           NEW.tsvectors = (
             setweight(to_tsvector('english', NEW.name), 'A') ||
             setweight(to_tsvector('english', NEW.brand), 'B')
           );
         END IF;
    END IF;
    RETURN NEW;
```

```
END $$
LANGUAGE plpgsql;

CREATE TRIGGER productSearchUpdate
BEFORE INSERT OR UPDATE ON Product
FOR EACH ROW
EXECUTE PROCEDURE productSearchUpdate();

CREATE INDEX search_idx ON Product USING GIN (tsvectors);

--------------------------------------------------------
-- TRIGGERS
--------------------------------------------------------

-- TRIGGER TO UPDATE THE RATING OF A PRODUCT WHEN A REVIEW IS ADDED OR CHANGED --
CREATE FUNCTION updateProductRating() RETURNS TRIGGER AS
$BODY$
BEGIN
    Update Product
    Set rating = (SELECT avg(review.rating) from Review where id_Product = NEW.id_Product)
    where Product.id = NEW.id_Product;
    RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER updateProductRating
AFTER INSERT OR UPDATE
ON Review
FOR EACH ROW
EXECUTE PROCEDURE updateProductRating();


-- TRIGGER TO PREVENT A USER FROM ADDING MORE ITEMS TO THE CART THAN THE AVAILABLE STOCK
CREATE FUNCTION verificaStock() RETURNS TRIGGER AS
$BODY$
BEGIN
IF NOT EXISTS(
    SELECT * from Product
    WHERE NEW.id_Product = Product.id
        AND NEW.quantity <= Product.stock
)
THEN RAISE EXCEPTION 'You can NOT add that much quantity to the cart';
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER verificaStock
BEFORE INSERT ON CartProduct
FOR EACH ROW
```

```
EXECUTE PROCEDURE verificaStock();


-- TRIGGER TO BLOCK BANNED USERS FROM WRITING REVIEWS
CREATE FUNCTION blockBannedUsers() RETURNS TRIGGER AS
$BODY$
BEGIN
IF NOT EXISTS(
    SELECT * from Customer
    WHERE Customer.id = New.id_Customer
        AND Customer.id in (
            SELECT id
            from account
            where isBanned = false
        )
)
THEN RAISE EXCEPTION 'You can NOT write reviews as you´ve been banned from the website';
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER blockBannedUsers
AFTER INSERT ON Review
FOR EACH ROW
EXECUTE PROCEDURE blockBannedUsers();


-- TRIGGER TO SEND A NOTIFICATION TO THE CUSTOMER WHEN THERE IS A CHANGE OF STATUS IN THEIR ORDER
CREATE FUNCTION orderStatusNotification() RETURNS TRIGGER AS
$BODY$
BEGIN
IF EXISTS (
    SELECT * from Customer
    where New.id_Customer = Customer.id
)
THEN INSERT INTO Notification(content, id_Customer) VALUES ('Your order status has been updated' , New.id_Customer);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER orderStatusNotification
AFTER UPDATE ON Purchase
FOR EACH ROW
EXECUTE PROCEDURE orderStatusNotification();
```

A.2. Database population

```
------------------------------------------------------ ACCOUNT ------------------------------------------------------
INSERT INTO ACCOUNT (id, username, password, email) VALUES (1, 'up201907716', '{a!.z27*NL-d$J7M', 'up201907716@up.pt');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (2, 'up201905497', '_Cw<n2wPmBVkL\gj', 'up201905497@up.pt');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (3, 'up201905477', 'p8%A6mGe@(]*D:GT', 'up201905477@up.pt');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (4, 'up201800700', 'Wb%hQD,6uB$/xc6D', 'up201800700@up.pt');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (5, 'firstUser', 'ZC73\Trx5/N$L7', 'vova10000@bukan.es');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (6, 'Pax Brandom', 'paxa213841' ,'paxa949494@barretodrums.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (7, 'Mark Peterson', 'iammark@2312', 'iammarkp@uioct.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (8, 'Lena Maxwell' , 'lenamax2138@pwd', 'lenamax@uioct.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (9, 'Evelin Watson', 'evelynwat321', 'evelinWatson@pickuplanet.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (10, 'Jagger Smith', 'jAGGERsmth', 'jaggerSmith@pickuplanet.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (11, 'Gage Hubbard', 'hubbard88213', 'GageHubbard@pesssink.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (12, 'Maeve Schwartz', 'maesch3952', 'Maeve Schwartz@shirtsthatshouldexist.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (13, 'Alexia Dickerson', 'ALEXDICKesrson@@,', 'alexiaDickerson@oanghika.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (14, 'Armani Fischer', 'aRmAnIfish', 'armaniFischer54@furnitt.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (15, 'Frank Curry', 'frankkcur2', 'frankCurry99@furnitt.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (16, 'Amare Burnett', 'amrBrunette8', 'amareBurnett76@pesssink.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (17, 'Jakayla Mercer', 'jakylaMercer22', 'JakaylaMercer12@oanghika.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (18, 'Marvin Keller', 'Mrvinkeller', 'MarvinKeller55@furnitt.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (19, 'Destiny Jacobs', 'jacobDstny', 'DestinyJacobs2@shirtsthatshouldexist.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (20, 'Quinten Crosby', 'QuentinCosby212', 'QuintenCrosby22@pesssink.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (21, 'Darnell Drake', 'darnellgodsPlan', 'DarnellDrake69@oanghika.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (22, 'Will Larson', 'willlarrrsun2', 'WillLarson23@shirtsthatshouldexist.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (23, 'Odin Harrell', 'odinodinharell', 'OdinHarrel1992@shirtsthatshouldexist.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (24, 'Rihanna Rosales', 'rosaumbrella23>', 'RihannaRosales1990@pesssink.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (25, 'Gideon Ruiz', 'ruizzz*-*', 'GideonRuiz2002@shirtsthatshouldexist.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (26, 'Gianni Potts', 'gianniPotts+', 'GianniPotts42@oanghika.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (27, 'Niko Valencia', 'nikoMadrid3<', 'NikoValencia97@oanghika.es');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (28, 'Gary Odom', 'garrryOdomm2', 'GaryOdom33@furnitt.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (29, 'Camren Spears', 'camrenBritney\', 'CamrenSpears88@furnitt.com');
INSERT INTO ACCOUNT (id, username, password, email) VALUES (30, 'Mohamed Kaiser', 'mohamedKais-', 'MohamedKaiser123@pesssink.com');
INSERT INTO ACCOUNT (id, username, password, email, isBanned) VALUES (31, 'Banned user', 'Banneduser2-', 'Banneduser@pesssink.com', True);

------------- ADMIN -------------
INSERT INTO ADMIN (id) VALUES (1);
INSERT INTO ADMIN (id) VALUES (2);
INSERT INTO ADMIN (id) VALUES (3);
INSERT INTO ADMIN (id) VALUES (4);

-------------- Cart ---------------
INSERT INTO Cart (id) VALUES (5);
INSERT INTO Cart (id) VALUES (6);
INSERT INTO Cart (id) VALUES (7);
INSERT INTO Cart (id) VALUES (8);
INSERT INTO Cart (id) VALUES (9);
INSERT INTO Cart (id) VALUES (10);
INSERT INTO Cart (id) VALUES (11);
INSERT INTO Cart (id) VALUES (12);
INSERT INTO Cart (id) VALUES (13);
INSERT INTO Cart (id) VALUES (14);
INSERT INTO Cart (id) VALUES (15);
INSERT INTO Cart (id) VALUES (16);
INSERT INTO Cart (id) VALUES (17);
INSERT INTO Cart (id) VALUES (18);
INSERT INTO Cart (id) VALUES (19);
INSERT INTO Cart (id) VALUES (20);
INSERT INTO Cart (id) VALUES (21);
INSERT INTO Cart (id) VALUES (22);
INSERT INTO Cart (id) VALUES (23);
INSERT INTO Cart (id) VALUES (24);
INSERT INTO Cart (id) VALUES (25);
INSERT INTO Cart (id) VALUES (26);
INSERT INTO Cart (id) VALUES (27);
INSERT INTO Cart (id) VALUES (28);
INSERT INTO Cart (id) VALUES (29);
INSERT INTO Cart (id) VALUES (30);
INSERT INTO Cart (id) VALUES (31);


------------------ CUSTOMER --------------------
INSERT INTO CUSTOMER (id, id_Cart) VALUES (5, 5);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (6, 6);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (7, 7);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (8, 8);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (9, 9);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (10, 10);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (11, 11);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (12, 12);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (13, 13);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (14, 14);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (15, 15);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (16, 16);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (17, 17);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (18, 18);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (19, 19);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (20, 20);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (21, 21);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (22, 22);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (23, 23);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (24, 24);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (25, 25);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (26, 26);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (27, 27);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (28, 28);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (29, 29);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (30, 30);
INSERT INTO CUSTOMER (id, id_Cart) VALUES (31, 31);

----------------------------------------------------------------------------------
---------------------------- CPU -------------------------------------------------
-----------------------------------------------------------
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (10, 'AMD Ryzen 9 5950X', 836.99, 'AMD', 5,'images/10.jpg' , 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (11, 'INTEL-Core i9-9900K', 571.99, 'Intel', 3,'images/11.jpg' , 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (12, 'INTEL Core i5-10600K', 247.99, 'Intel', 5, 'images/12.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (13, 'AMD Ryzen 7 5800X', 449.99, 'AMD', 4, 'images/13.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (14, 'INTEL Celeron G5925', 54.99, 'Intel', 6, 'images/14.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (15, 'AMD Ryzen Threadripper 3990X', 4299.99, 'AMD', 3, 'images/15.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (16,
'INTEL Xeon Gold 6248', 3739.00, 'Intel', 2, 'images/16.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (17, 'AMD Athlon 240GE', 118.63, 'AMD', 1, 'images/17.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (18, 'INTEL-Core i5-9600K', 188.10, 'Intel', 0, 'images/18.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (19, 'AMD Ryzen 5 3600X', 245.90, 'AMD', 4, 'images/19.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');

INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (10, 3.4, 4.9, 'AM4', 32, 16);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (11, 3.6, 5, 'LGA1151', 16, 8);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (12, 4.1, 4.8, 'LGA1200', 12, 6);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (13, 3.8, 4.7, 'AM4', 16, 8);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (14, 3.6, 0, 'LGA1200', 2, 2);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (15, 2.9, 4.3, 'TRX4', 128, 64);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (16, 2.5, 3.9, 'LGA3647', 40, 20);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (17, 3.5, 0, 'AM4', 4, 2);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (18, 3.7, 4.6, 'LGA1151', 6, 6);
INSERT INTO CPU(id, baseFreq, turboFreq, socket, threads, cores) VALUES (19, 3.8, 4.4, 'AM4', 12, 6);


-------------------------------------------------------------------------------------------------------------- GPU ----------------------------------------------------------------------------------------------------------
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (20, 'ASUS GeForce GT 1030', 115.72, 'ASUS', 15, 'images/20.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (21, 'ASUS ROG STRIX GeForce RTX 3080 Ti', 1899.00, 'ASUS', 2, 'images/21.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (22, 'MSI GeForce GT 710', 78.16, 'MSI', 6, 'images/22.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (23,
'AMD Radeon Pro WX 3100', 299.14, 'AMD', 6, 'images/23.jpg', 'Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus
ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (24,
'AMD Raden Pro WX 2100', 205.62, 'AMD', 3, 'images/24.jpg', 'Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut,
tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (25,
'HP NVIDIA Tesla K20X', 14421.50, 'HP', 1, 'images/25.jpg', 'Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut,
tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (26,
'XFX Radeon RX 6600 XT', 899.90, 'XFX', 2, 'images/26.jpg', 'Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut,
tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (27,
'Gigabyte Radeon RX 6900 XT', 1469.90, 'Gigabyte', 8, 'images/27.jpg', 'Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare
vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (28,
'EVGA GeForce RTX 3080 FTW', 1499.90, 'EVGA', 7, 'images/28.jpg', 'Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel
faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (29,
'Zotac Gaming GeForce RTX 3060 Ti', 1099.00, 'Zotac', 4, 'images/29.jpg', 'Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare
vel faucibus ut, tempor at ipsum. Nam elementum');


INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(20, 2, 1252, 1531, 1, 0);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(21, 12, 1365, 1845, 2, 3);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(22, 2, 954, 954, 1, 0);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(23, 4, 925, 1219, 0, 1);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(24, 2, 925, 1219, 0, 1);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(25, 6, 732, 732, 0, 0);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(26, 8, 2092, 2589, 1, 3);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(27, 16, 2050, 2285, 1, 2);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(28, 10, 1440, 1800, 1, 3);
INSERT INTO GPU(id, memory, coreClock, boostClock, hdmiPorts, displayPorts) VALUES
(29, 8, 1410, 1695, 1, 3);


----------------------------------------------------------------------------------
---------------------------- Motherboard -----------------------------------------
-------------------------------------------------------------------
```

```
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (30, 'ASUS TUF Z390-Plus Gaming', 234.81, 'ASUS', 4, 'images/30.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (31, 'ASUS ROG STRIX Z590-F GAMING WIFI', 319.90, 'ASUS', 2, 'images/31.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (32, 'MSI MPG X570 Gaming Edge', 221.32, 'MSI', 3, 'images/32.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (33, 'MSI MPG X570 Gaming Plus', 185.29, 'MSI', 6, 'images/33.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (34, 'ASUS ROG MAXIMUS XIII EXTREME GLACIAL', 1717.38, 'ASUS', 1, 'images/34.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (35, 'SUPERMICRO X11SBA-LN4F', 355.72, 'SUPERMICRO', 0, 'images/35.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (36, 'ASUS P10S-X', 365.04, 'ASUS', 0, 'images/36.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (37, 'ATX ASRock TRX40 Taichi', 499.90, 'ASRock', 4, 'images/37.jpg',  'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (38, 'Gigabyte B450 Gaming X', 75.90, 'Gigabyte', 7, 'images/38.jpg',  'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (39, 'Asus ROG Maximus XII Apex', 495.90, 'ASUS', 3, 'images/39.png',  'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');


INSERT INTO Motherboard(id, socket, type) VALUES (30, 'LGA1151', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (31, 'LGA1200', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (32, 'AM4', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (33, 'AM4', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (34, 'LGA1200', 'EATX');
INSERT INTO Motherboard(id, socket, type) VALUES (35, 'CPU Onboard - Intel Pentium', 'MINI-ITX');
INSERT INTO Motherboard(id, socket, type) VALUES (36, 'LGA1151', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (37, 'sTRX4', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (38, 'AM4', 'ATX');
INSERT INTO Motherboard(id, socket, type) VALUES (39, 'LGA1200', 'ATX');
```

```
----------------------------------------------------------------------------------
---------------------------- Storage ---------------------------------------------
---------------------------------------------------------------
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (40, 'DDR3 KINGSTON KVR16N11/8', 50.99, 'KINGSTON', 4, 'images/40.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (41, 'DDR4 CISCO UCS-MR-1X322RV-A=', 2727.15, 'CISCO', 5, 'images/41.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (42, 'DDR2 NILOX NXS1800H1C6', 8.99, 'NILOX', 0, 'images/42.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (43, 'DDR4 G.SKILL F4-2133C15Q-32GVR', 180.29, 'GSKILL', 21, 'images/43.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (44, 'DDR4 CORSAIR CMK128GX4M8A2666C16', 780.74, 'CORSAIR', 12, 'images/44.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (45, 'Samsung 970 Evo Plus', 139.99, 'Samsung', 31, 'images/45.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (46, 'Kingston A400', 27.99, 'KINGSTON', 4, 'images/46.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (47, 'Western Digital Caviar Blue', 30.99, 'Western Digital', 7, 'images/47.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (48, 'Seagate Barracuda Compute', 149.99, 'Seagate', 8, 'images/48.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (49, 'Samsung 980 Pro', 79.98, 'Samsung', 8, 'images/49.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');

INSERT INTO Storage(id, type, capacity) VALUES (40, 'RAM', 8);
INSERT INTO Storage(id, type, capacity) VALUES (41, 'RAM', 32);
INSERT INTO Storage(id, type, capacity) VALUES (42, 'RAM', 1);
INSERT INTO Storage(id, type, capacity) VALUES (43, 'RAM', 32);
INSERT INTO Storage(id, type, capacity) VALUES (44, 'RAM', 128);
INSERT INTO Storage(id, type, capacity) VALUES (45, 'M.2', 1000);
INSERT INTO Storage(id, type, capacity) VALUES (46, 'SSD', 240);
INSERT INTO Storage(id, type, capacity) VALUES (47, 'HDD', 500);
INSERT INTO Storage(id, type, capacity) VALUES (48, 'HDD', 8000);
INSERT INTO Storage(id, type, capacity) VALUES (49, 'M.2', 250);


----------------------------------------------------------------------------------
---------------------------- PcCase ----------------------------------------------
--------------------------------------------------------------
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (50, 'Corsair 4000D Airflow', 79.99, 'CORSAIR', '453mm x 230mm x 466mm', 7,
'images/50.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (51, 'Lian Li PC-O11 Dynamic', 149.99, 'Lian Li', '445mm x 272mm x 446mm',
4, 'images/51.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (52, 'Lian Li O11 Dynamic Mini Snow Edition', 129.90, 'Lian Li', '420mm x
269.5mm x 380mm', 2, 'images/52.jpg', 'Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum.
Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (53, 'Cooler Master MasterBox NR200P', 135.98, 'Cooler Master', '376mm x
185mm x 292mm', 21, 'images/53.jpg', 'Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum.
Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (54, 'Thermaltake Level 20', 999.99, 'Thermaltake', '732mm x 280mm x
688mm', 3, 'images/54.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam
elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (55, 'Jonsbo TR03-A', 507.00, 'Jonsbo', '673mm x 238mm x 595mm', 0,
'images/55.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (56, 'Corsair 760T', 417.50, 'CORSAIR', '564mm x 246mm x 568mm', 0,
'images/56.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (57, 'Corsair iCUE 4000X RGB', 114.89, 'CORSAIR', '453mm x 230mm x 466mm',
12, 'images/57.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (58, 'NZXT H510', 79.90, 'NZXT', '428mm x 210mm x 460mm', 6,
'images/58.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size, stock, image, description)
VALUES (59, 'NZXT H710', 149.90,'NZXT', '494mm x 230mm x 516mm', 7,
'images/58.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');

INSERT INTO PcCase(id, type, weight, color) VALUES (50, 'EATX', 7.8, 'Black');
INSERT INTO PcCase(id, type, weight, color) VALUES (51, 'EATX', 11.9, 'Black');
INSERT INTO PcCase(id, type, weight, color) VALUES (52, 'ATX', 9, 'White');
INSERT INTO PcCase(id, type, weight, color) VALUES (53, 'MINI-ITX', 5, 'Orange');
INSERT INTO PcCase(id, type, weight, color) VALUES (54, 'EATX', 32, 'Black');
INSERT INTO PcCase(id, type, weight, color) VALUES (55, 'ATX', 15, 'Silver');
INSERT INTO PcCase(id, type, weight, color) VALUES (56, 'EATX', 11.15, 'Black');
INSERT INTO PcCase(id, type, weight, color) VALUES (57, 'ATX', 7.85, 'Black');
INSERT INTO PcCase(id, type, weight, color) VALUES (58, 'ATX', 6.6, 'White');
INSERT INTO PcCase(id, type, weight, color) VALUES (59, 'EATX', 12.1, 'White');

----------------------------------------------------------------------------------
---------------------------- Cooler ----------------------------------------------
--------------------------------------------------------------
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (60,
'Cooler Master Hyper 212 RGB Black Edition', 44.99, 'Cooler Master', 8,
'images/60.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size,  stock, image, description)
VALUES (61, 'Corsair iCUE H100i ELITE CAPELLIX', 109.99, 'Corsair', '240mm', 4,
'images/61.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus
tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (62,
'be quiet! Dark Rock Pro 4', 89.90, 'be quiet!', 12,  'images/62.jpg', 'Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare
vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (63,
'Deepcool ASSASSIN II', 264.30, 'Deepcool', 2, 'images/63.jpg', 'Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus
ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (64,
'Thermaltake Engine 27', 69.99, 'Thermaltake', 2, 'images/64.jpg', 'Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel
faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (65,
'be quiet! Dark Rock TF', 135.55, 'be quiet!', 3, 'images/65.jpg', 'Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel
faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (66,
'Corsair A500', 92.17, 'Corsair' , 7, 'images/66.jpg', 'Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut,
tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size,  stock, image, description)
VALUES (67, 'EK AIO Basic 360', 128.99, 'EK', '360mm', 3, 'images/67.jpg', 'Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare
vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size,  stock, image, description)
VALUES (68, 'Corsair iCUE H150i RGB Elite Capellix', 184.90, 'Corsair', '360mm',
5,'images/68.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, size,  stock, image, description)
VALUES (69, 'Mars Gaming ML120 RGB', 54.90, 'Mars Gaming', '120mm',
12,'images/69.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');


INSERT INTO Cooler(id, type) VALUES (60, 'Air');
INSERT INTO Cooler(id, type) VALUES (61, 'Water');
INSERT INTO Cooler(id, type) VALUES (62, 'Air');
INSERT INTO Cooler(id, type) VALUES (63, 'Air');
INSERT INTO Cooler(id, type) VALUES (64, 'Air');
INSERT INTO Cooler(id, type) VALUES (65, 'Air');
INSERT INTO Cooler(id, type) VALUES (66, 'Air');
INSERT INTO Cooler(id, type) VALUES (67, 'Water');
INSERT INTO Cooler(id, type) VALUES (68, 'Water');
INSERT INTO Cooler(id, type) VALUES (69, 'Water');


----------------------------------------------------------------------------------
---------------------------- PowerSupply -----------------------------------------
-------------------------------------------------------------------
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (70, 'Corsair RM (2019)', 114.99, 'Corsair', 5, 'images/70.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (71, 'EVGA BQ', 53.42, 'EVGA',  3, 'images/71.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (72, 'Cooler Master V SFX', 144.99, 'Cooler Master', 1, 'images/72.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (73, 'Corsair RMx (2018)', 109.99, 'Corsair', 4, 'images/73.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (74, 'MSI MPG A-GF', 69.98, 'MSI', 2, 'images/74.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (75, 'be quiet! Straight Power 11', 89.90, 'be quiet!', 0, 'images/75.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (76, 'be quiet! Pure Power 11', 137.00, 'be quiet!', 1, 'images/76.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (77, 'Seasonic Core GC', 79.90, 'Seasonic', 3, 'images/77.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (78, 'Nox Urano VX', 60.90, 'Nox', 7, 'images/78.jpg', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO Product(id, name, price, brand, stock, image, description) VALUES (79, 'EVGA SuperNOVA GT', 69.90, 'EVGA', 6, 'images/79.jpg' , 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus risus, ornare vel faucibus ut, tempor at ipsum. Nam elementum');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (70, 750, '80+
Gold', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (71, 600, '80+
Bronze', 'Semi-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (72, 850, '80+
Gold', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (73, 750, '80+
Gold', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (74, 650, '80+
Gold', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (75, 750, '80+
Gold', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (76, 300, '80+
Bronze', 'Non-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (77, 650, '80+
Gold', 'Non-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (78, 750, '80+
Bronze', 'Full-Modular');
INSERT INTO PowerSupply(id, wattage, certification, type) VALUES (79, 650, '80+
Gold', 'Full-Modular');

----------------------------------------------------------------------------------
-------------------------------- Other -------------------------------------------
--------------------------------------------------------------------------------


----------------------------------------------------------------------------------
---------------------------- Address ---------------------------------------------
---------------------------------------------------------------
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (5, 'Rua Costa'
, 108, '4795-158');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (6, 'Rua Nossa
Senhora Fátima', 95, '3330-078');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
VALUES (7, 'Rua Capitão Henrique Galvão', 94, '2705-210', 3, 'Esquerdo');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
VALUES (8, 'Rua Caminho Cruz' , 102, '4450-540', 7, 'Trás');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (9, 'Rua Germana
Tânger' , 123, '2725-239');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (10, 'Rua
Cruzes', 72, '4750-791');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (11, 'Rua Poeta
João Ruiz', 151, '6225-259');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (12, 'Rua Parque
Gondarim', 82, '4405-747');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
VALUES (13, 'Praceta Conde Arnoso', 102, '2640-097', 2, 'Frente');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
VALUES (14, 'Rua Regato ', 83, '6090-025', 0, 'Direito');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (15, 'Rua Vale
Formoso', 18, '8000-426');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
VALUES (16, 'Rua Projetada', 108, '2900-593', 6, 'Trás');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber)
```

```
VALUES (17, 'Rua São Gonçalo', 1052, '7830-374', 1, 'Esquerdo');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (18, 'Avenida Boavista', 108, '4920-100');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (19, 'Rua Cabral Antunes', 69, '3750-721');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (20, 'Avenida Boavista', 108, '4920-100');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (21, 'Rua Heróis Ultramar', 56, '2205-623');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber) VALUES (22, 'Rua Doutor Alfredo Freitas', 128, '3700-383', 3, 'Frente');
INSERT INTO Address(id, streetName, streetNumber, zipcode) VALUES (23, 'Rua Rainha Santa Isabel', 1044, '3250-265');
INSERT INTO Address(id, streetName, streetNumber, zipcode, floor, aptNumber) VALUES (24, 'Rua Cimo Vila', 125, '4590-083', 1, 'Esquerdo');


----------------------------------------------------------------------------------
---------------------------- Purchase --------------------------------------------
----------------------------------------------------------------
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (1, '2021/11/28', '2021/12/4', 'Processing', 5, 5, 5, 5);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (2, '2021/11/28', '2021/12/2', 'Packed', 7, 7, 7, 7);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (3, '2021/11/28', '2021/12/7', 'Processing', 9, 9, 9, 9);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (4, '2021/11/21', '2021/12/23', 'Delivered', 18, 18, 18, 18);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (5, '2021/11/26', '2021/12/29', 'Delivered', 23, 23, 23, 23);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (6, '2021/11/16', '2021/11/19', 'Delivered', 24, 24, 24, 24);
INSERT INTO Purchase(id, orderDate, deliveryDate, orderStatus, id_Customer, id_Address, id_PaymentMethod, id_Cart) VALUES (7, '2021/11/27', '2021/11/30', 'Shipped', 12, 12, 12, 12);

------------- PaymentMethod -------------
INSERT INTO PaymentMethod(id) VALUES (5);
INSERT INTO PaymentMethod(id) VALUES (6);
INSERT INTO PaymentMethod(id) VALUES (7);
INSERT INTO PaymentMethod(id) VALUES (8);
INSERT INTO PaymentMethod(id) VALUES (9);
INSERT INTO PaymentMethod(id) VALUES (12);
INSERT INTO PaymentMethod(id) VALUES (18);
INSERT INTO PaymentMethod(id) VALUES (20);
INSERT INTO PaymentMethod(id) VALUES (22);
INSERT INTO PaymentMethod(id) VALUES (23);
INSERT INTO PaymentMethod(id) VALUES (24);
```

```sql
INSERT INTO PaymentMethod(id) VALUES (25);
INSERT INTO PaymentMethod(id) VALUES (29);
INSERT INTO PaymentMethod(id) VALUES (30);
INSERT INTO PaymentMethod(id) VALUES (31);

--------------------------- Paypal ---------------------------
INSERT INTO Paypal (id, email) VALUES (5, 'vova10000@bukan.es');
INSERT INTO Paypal (id, email) VALUES (6, 'paxa949494@barretodrums.com');
INSERT INTO Paypal (id, email) VALUES (7, 'iammarkp@uioct.com');
INSERT INTO Paypal (id, email) VALUES (8, 'lenamax@uioct.com');
INSERT INTO Paypal (id, email) VALUES (9, 'evelinWatson@pickuplanet.com');
INSERT INTO Paypal (id, email) VALUES (12, 'Maeve Schwartz@shirtsthatshouldexist.com');
INSERT INTO Paypal (id, email) VALUES (29, 'CamrenSpears88@furnitt.com');
INSERT INTO Paypal (id, email) VALUES (30, 'MohamedKaiser123@pesssink.com');

--------------------------- Card ---------------------------
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (18, 'Marvin Keller', 5557028833413930, '2022/12/01', 752);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (20, 'Quinten Crosby', 4532809664584094, '2024/11/01', 684);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (22, 'Will Larson', 5421946123168068, '2023/12/01', 491);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (23, 'Odin Harrell', 5118045101110610, '2025/02/01', 892);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (24, 'Rihanna Rosales', 4485749073456673, '2024/01/01', 515);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (25, 'Gideon Ruiz', 4485488722909754, '2024/04/01', 539);
INSERT INTO Card (id, name, number, expDate, cvv) VALUES (31, 'George Washington', 5135825255955624, '2023/02/01', 451);

------------------------ CustomerAddress -------------------------
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (5, 5);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (6, 6);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (7, 7);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (8, 8);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (9, 9);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (10, 10);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (11, 11);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (12, 12);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (13, 13);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (14, 14);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (15, 15);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (16, 16);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (17, 17);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (18, 18);
INSERT INTO CustomerAddress (id_Customer, id_Address) VALUES (19, 19);

------------------------------ CartProduct ------------------------------
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (5, 10, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (7, 21, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (7, 11, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (9, 31, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (9, 62, 2);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (18, 73, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (23, 77, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (25, 12, 1);
INSERT INTO CartProduct(id_Cart, id_Product, quantity) VALUES (29, 44, 3);

------------------------ Wishlist --------------------------
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (6, 14);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (6, 17);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (6, 24);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (7, 10);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (7, 11);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (7, 64);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (8, 21);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (8, 27);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (8, 55);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (9, 54);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (10, 37);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (12, 79);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (14, 69);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (17, 11);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (19, 12);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (21, 34);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (23, 21);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (23, 33);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (24, 24);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (25, 18);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (27, 44);
INSERT INTO Wishlist(id_Customer, id_Product) VALUES (29, 77);


---------------------------------------------Review ----------------------------------------------
INSERT INTO Review(id, text, rating, id_Customer, id_Product) VALUES (1, 'Very nice', 4, 6, 12);
INSERT INTO Review(id, text, rating, id_Customer, id_Product) VALUES (2, 'Very bad', 1, 15, 14);
INSERT INTO Review(id, text, rating, id_Customer, id_Product) VALUES (3, 'Very bad', 1, 5, 12);
```

Revision history

Changes made to the first submission:

1. A4 and A5 added;
2. Indexes and Triggers added;
3. Transaction added.

GROUP2175, 01/12/2021

- Carlos Veríssimo, up201907716@edu.fe.up.pt (Editor)
- Duarte Sardão, up201905497@edu.fe.up.pt
- Nuno Jesus, up201905477@edu.fe.up.pt
- Tomás Torres, up201800700@edu.fe.up.pt