

Compte-Rendu SQL SAE3 equipe 1-1 :

Ce project a été menée par:

- Titouan Gautier
- Maxime Santos Pereira
- Arthur Hamelin
- Luna Manson

1 - Schéma relationnel :





On voit ci-dessus les différentes tables présente dans la base de donnée :

- **Sport** qui contient les noms des 4 sports présent sur notre site (Football, Volleyball, Badminton et Arts-Martiaux), la clé primaire de cette table étant **nusport**.
- **Product** qui contient les différents produits du site avec comme clé primaire **idproduct** et comme clé étrangère **nusport** qui relie à la table **Sport (nusport)**.
- **Catalog** qui contient les différentes tailles, couleurs et quantités des produits, la clé primaire étant l'**id** de la variante du produit. Et la clé étrangère étant **nuproduct** liant à la table **Product (idproduct)**.

- **Cart** qui contient les différents paniers enregistrer par les utilisateurs, elle a pour clé primaire **id** qui correspond à un produit présent dans un panier d'un utilisateur et a pour clé étrangère **iduser** reliant à la table **User (id)** et la clé étrangère **idvariant** reliant à la table **Catalog (id)**.
- **Order** qui contient les commandes des utilisateurs, la clé primaire est **id** et les clés étrangères sont **iduser** qui relie à la table **User (id)** et **idlocation** qui relie à la table **Location (idlocation)**.
- **Order_products** qui contient les produits des différentes commandes des utilisateurs. Les clés primaires sont **idorder**, **idproduct** et **idvariant**, ces clés sont également étrangères permettant d'être reliée à **Order (id)**, **Product (idproduct)** et **Catalog (id)**.
- **User** qui contient les différents utilisateurs avec leurs informations de connexion. La clé primaire est **id**.
- **Location** qui contient les différentes adresses de l'utilisateur. La clé primaire est **idlocation** et la clé étrangère est **id** qui relie à la table **User (id)**.
- **Information** qui contient les différentes coordonnées de l'utilisateur. La clé primaire est **id**, elle est également étrangère est relie à la table **User (id)**.
- **Captcha** qui regroupe les différents captcha généré lors de la création de compte. La clé primaire est **captcha_id**.
- **VerifKey** qui contient les différentes clés générées lorsqu'on souhaite une vérification supplémentaire par mail. La clé primaire est **id** et la clé étrangère est **idUser** reliant à la table **User (id)**.

Une dernière table a été mise en place **Stock_alerte** qui permet de stocker les id des variants ayant une rupture de stock.

 stock_alerte	
 id	int(11)

2 - Contraintes :

Les contraintes suivantes ont été réaliser lors de la création des tables.

Il y a une contrainte **status_not_valid** sur la table **User** pour vérifier si le Status est dans le bon format :

```
constraint status_not_valid
    check(status in ('Administrateur','Client','Caspor')),
```

Il y a deux contraintes sur la table **Product**, la première **type_not_valid** pour vérifier si le type est dans le bon format, et la deuxième **gender_not_valid** pour vérifier si le genre du produit est dans le bon format :

```
constraint type_not_valid
    check( type in ('Vêtement','Chaussure','Equipement') ),
constraint gender_not_valid
    check( gender in ('Homme','Femme','Mixte'))
```

Il y a une contrainte **status_not_valid** sur la table **Order** pour vérifier si le Status est dans le bon format :

```
constraint status_not_valid
    check(state in ('Non preparer', 'En
preparation', 'Preparer', 'Expedier'))
```

De plus 3 triggers ont été réalisés, **dateLastUpdate** qui permet de mettre à jour la date de la dernière mise à jour si le status a été modifier. **TimeIsUp** qui vérifie que la différence entre la nouvelle date de dernière mise à jour et la date actuelle n'est pas supérieur ou égale à 2 mois (60 jours) si c'est le cas les adresses modifiées sont supprimer de la table Location et de la table Order. Et **StockNotAvailable** qui vérifie lors d'une modification sur la table catalog que le variant modifié a soit une quantité nulle ce qui entraine l'ajout de son id à la table Stock_alerte soit une quantité différent de nulle ce qui entraine le retrait de son id de la table stock_alerte si il est présent.

```
create or replace trigger TimeIsUp after update on location for each row
begin
    if datediff(NEW.dateLastUpdate, CURRENT_DATE) >= 60 then
        delete from location where idlocation = NEW.idlocation;
        delete from `order` where `order`.idlocation = NEW.idlocation;
    end if;
end;

create or replace trigger dateLastUpdate before update on user for each row
begin
    if old.status != new.status then
        SET new.dateLastUpdate = NOW();
    end if;
end;

create or replace trigger StockNotAvailable after update on catalog for
each row
begin
    if (new.quantity = 0) then
        insert into stock_alerte values (new.id);
    end if;
    if (new.id in (select s.id from stock_alerte s) and new.quantity !=
0) then
        delete from stock_alerte where id = new.id;
    end if;
end;
```

3 - Package

Package sport

Nom	Paramètre entrant	Paramètres sortant	Commentaires
getNameSport	nusport int	Nom du sport qui est associé à nusport	Permet d'avoir le nom du sport en fonction de son id
getIdSport	name varchar	Id du sport associé à name	Permet d'avoir l'id du sport en fonction de son nom
getAll		Toutes les lignes de la table sport	Permet de récupérer toutes les informations sur les sports

Package user

Nom	Paramètre entrant	Paramètres sortant	Commentaires
verifyLogin	loginSearch varchar	login varchar	Permet de vérifier si un login donnée est présent dans la BD
verifyEmail	mailSearch	login varchar	Permet de vérifier si un mail donnée est présent dans la BD
verifyPhone	phone int	login varchar	Permet de vérifier si un Numéro de téléphone donnée est présent dans la BD
verifyId	newId int	login varchar	Permet de vérifier si un ID donnée est présent dans la BD
getUserByLogin	loginSearch varchar	login varchar, id int	Permet de récupérer un user par son login

Nom	Paramètre entrant	Paramètres sortant	Commentaires
getUserById	idSearch varchar	id int, login varchar, cookieId varchar, status varchar, isVerified tinyint(1), isALive tinyint(1)	Permet de récupérer un user par son ID
getUserByEmail	mailSearch varchar	login varchar, id int	Permet de récupérer un user par son adresse mail
getPasswordById	idSearch int	password varchar, salt varchar	Permet de récupérer un mot de passe par un ID
getStatusById	idSearch int	status varchar	Permet de récupérer un status par un ID
getUserInfoById	idUser int	Toutes les lignes de Information où l'id est égale à iduser	Permet de récupérer les information d'un user à partir de son id
getUserLocationById	idUser int	id int	Permet de récupérer la Location d'un user avec son ID
loginMail	mail varchar	password varchar	Permet de récupérer un mot de passe par une adresse mail
addUser	newid int, newlogin varchar, newpass varchar, newsalt varchar, newcookie varchar, newstatus varchar, newverif boolean, newalive boolean, newdate datetime		Permet d'ajouter un user

Nom	Paramètre entrant	Paramètres sortant	Commentaires
addInformation	newid int, newfirstname varchar, newname varchar, newmail varchar, newmobile int, newfix int		Permet d'ajouter les coordonnées d'un user par son ID
createUser	newId int, newLogin varchar, newPass varchar, newSalt varchar, newfirstname varchar, newname varchar, newEmail varchar, newMobile int, newFix int, dateLastUpdate date		Permet d'ajouter un user et ses coordonnées
delUser	iduser int		Permet de supprimer un user, ses coordonnées et sa location
setCookield	newCookield varchar(45), iduser int		Permet d'ajouter un cookie à un user
delCookield	iduser int		Permet de supprimer le cookie d'un user
updateCoordonnees	iduser int, newfirstname varchar(255), newname varchar(255), newmail varchar(255), newmobile int, newfix int		Permet de mettre à jour les différentes coordonnées d'un user
updateLocalisation	newidlocation int, iduser int, newname varchar(255) ,newlocation varchar(255), newcode int, newcity varchar(255), newdep varchar(255), newcountry varchar(255)		Permet de mettre à jour les différentes localisations d'un user
updateUtilisateur	iduser int, newlogin varchar(255), newpass varchar(255)		Permet de mettre à jour un user

Nom	Paramètre entrant	Paramètres sortant	Commentaires
updateStatus	idUser int, newstatus varchar(20)		Permet de mettre à jour le status d'un user
getLocationById	idloc integer	Toutes les lignes de Location où l'idlocation est égale à idloc	Permet de connaître la Location d'un user à partir de son id
updateLastName	targetId integer, newLastName varchar(255)		Permet de mettre à jour le nom de famille du User
updateFirstName	targetId integer, newFirstName varchar(255)		Permet de mettre à jour le prénom du user
updateEmail	targetId integer, newMail varchar(255)		Permet de mettre à jour l'email du user
updateMobile	targetId integer, newMobile varchar(255)		Permet de mettre à jour le numéros de téléphone portable du user
updateFixe	targetId integer, newFixe varchar(255)		Permet de mettre à jour le numéros de téléphone fixe du user
updatePassword	targetId integer, newPass varchar(255), newSalt varchar(45)		Permet de mettre à jour le password du user

Nom	Paramètre entrant	Paramètres sortant	Commentaires
verifySalt	newSalt varchar(255)	login varchar	Permet de vérifier que le sel n'est pas déjà présent dans la BD
getLocationByIdAndUserId	idUser int, idLoc int	Toutes les lignes de Location où l'id est égale à iduser et l'idlocation est égale à idloc	Permet de trouver une location d'un user
isUniqueAddressName	searchName varchar(255), searchIdUser int	count int	Permet de vérifier que le nom de l'adresse ne figure pas déjà dans les adresses du user
getAdresseById	searchId int	name varchar, location varchar, codepostal varchar, city varchar, department varchar, latitude double, longitude double, isDefault tinyint(1)	Permet de récupérer l'adresse avec l'id du user
verifyLocId	searchId int	name varchar(255)	Permet de récupérer le nom d'une location grace à son id

Nom	Paramètre entrant	Paramètres sortant	Commentaires
updateLocById	searchId int, newId int, idUser int, newLocName varchar(255), newLocAddress varchar(255), newLocCode int, newLocCity varchar(255), newLocDep varchar(255), newLocCountry varchar(255), newLocLat double, newLocLong double, newIsDefault bool, dateCreation date		Permet de mettre à jour une Location avec son id
addressIsDead	searchId int, newDateLastUpdate datetime		Permet de vérifier si une adresse est en morte
countAddressByIdAndName	searchId int, searchName varchar(255)	total int	Permet de compter le nombre d'adresse ayant le meme nom et le meme id que celui donné en paramètre
createLoc	newId int, idUser int, newLocName varchar(255), newLocAddress varchar(255), newLocCode int, newLocCity varchar(255), newLocDep varchar(255), newLocCountry varchar(255), newLocLat double, newLocLong double, newIsDefault bool, dateCreation date		Permet de créer une Location
sameAdresse	searchUserId int, searchAddress varchar(255), searchCity varchar(255)	total int	Permet de vérifier si deux adresses sont pareilles, renvoie le nombre d'adresses identiques

Nom	Paramètre entrant	Paramètres sortant	Commentaires
sameAdresseModif	searchUserId int, searchAddress varchar(255), searchCity varchar(255)	total int	Permet de vérifier si deux adresses sont pareilles, renvoie le nombre d'adresses identiques et l'id
countAliveAddressByUserId	searchUserId int	total int	Permet de compter le nombre d'adresses actives par user
getAllUser		Toutes les lignes de user	Permet de récupérer tous les users
getAllLocation		Toutes les lignes de Location	Permet de récupérer toutes les Locations
getAllInformation		Toutes les lignes de Information	Permet de récupérer toutes les Informations
getIsVerifiedById	idSearch VARCHAR(255)	isALive tinyint(1)	Permet de vérifier qu'un user est vérifié à partir de son id
getIsALiveById	idSearch VARCHAR(255)	isALive tinyint(1)	Permet de vérifier si un user est en vie avec son id

Nom	Paramètre entrant	Paramètres sortant	Commentaires
getDateLastUpdateById	idSearch VARCHAR(255)	dateLastUpdate datetime	Permet de récupérer la date du dernier update d'un user avec son id
userIsDead	searchId int, newDateLastUpdate date		Permet de mettre un user mort avec son id
setUserVerified	searchId int, newDate datetime		Permet de vérifier un user avec son id
changeStatus	searchId int, newStatus varchar(255)		Permet de changer le status de l'user
countUser		count int	Permet d'obtenir le nombre d'user
getUserByLocationId	searchId int	id int	Permet d'avoir l'utilisateur avec l'une des de ses adresses
getAllUserWithStep	start int, step int	id int, login varchar, status varchar, isVerified tinyint(1), isALive tinyint(1), firstname varchar, name varchar, mail varchar	Permet d'avoir un certain nombre d'utilisateur

Package product

Nom	Paramètre Entrant	Paramètre Sortant	Commentaire
-----	-------------------	-------------------	-------------

Nom	Paramètre Entrant	Paramètre Sortant	Commentaire
getProductBySport	sport integer	Toutes les lignes de product où nusport est égale à sport	Permet d'avoir les différents produits d'un sport spécifique
getProductByType	newtype varchar(15)	Toutes les lignes de product où type est égale à newtype	Permet d'avoir les différents produits d'un type spécifique
getProductByBrand	newbrand varchar(255)	Toutes les lignes de product où brand est égale à newbrand	Permet d'avoir les différents produits d'une marque spécifique
orderByPriceAsc		Toutes les lignes de product trier par prix croissant	Permet d'avoir les différents produits trié par prix croissant
orderByPriceDesc		Toutes les lignes de product trier par prix décroissant	Permet d'avoir les différents produits trié par prix décroissant
getProductBySize	size varchar(3)	Toutes les lignes de product où les produits ont size dans leur catalogue	Permet d'avoir les différents produits d'une taille spécifique

Nom	Paramètre Entrant	Paramètre Sortant	Commentaire
getProductByColor	color varchar(20)	Toutes les lignes de product où les produits ont color dans leur catalogue	Permet d'avoir les différents produits d'une couleur spécifique
getProductBySportType	sport integer, newtype varchar(15)	Toutes les lignes de product où nusport est égale à sport, type est égale à newtype et isALiva qui vaut true	Permet d'avoir les différents produits d'un sport et d'un type spécifique
getProductById	id integer	Toutes les lignes de product où idproduct est égale à id	Permet d'avoir un product par son ID
addProduct	newid int, newtype varchar(15), newnusport int, newmarque varchar(255), newnom varchar(255), newgenre varchar(5), newprix float, newdesc varchar(255), newimage text, newIsALive boolean, newDate datetime		Permet d'ajouter un product à la BD
updateProduct	id int, newtype varchar(15), newnusport int, newmarque varchar(255), newnom varchar(255), newgenre varchar(5), newprix float, newdesc varchar(255)		Permet de modifier un product
updatePrice	nuproduct int, newprice int		Permet de mettre à jour le prix d'un product

Nom	Paramètre Entrant	Paramètre Sortant	Commentaire
updateDescription	nuproduct int, newdesc varchar(255)		Permet de mettre à jour la description d'un product
updateImage	nuproduct int, newimage text		Permet de mettre à jour le chemin vers l'image d'un product
delProduct	nuproduct int		Permet de supprimer un product
getAll		Toutes les lignes de product	Permet de récupérer tout les produits
getAllAsAlive		Toutes les lignes de product où isALive vaut true	Permet de récupérer tout les produits en vie
getAllNotAlive		Toutes les lignes de product où isALive vaut false	Permet de récupérer tout les produits qui ne sont pas en vie.
getAllBrand		brand varchar(255)	Permet de récupérer toutes les marques
getProductByName	newname varchar(255)	Toutes les lignes de product où le name vaut newname	Permet de récupérer un produit selon son nom

Nom	Paramètre Entrant	Paramètre Sortant	Commentaire
getProductByNameWithoutSelf	newname varchar(255), id int	Toutes les lignes de product où name est égale au newname et idproduct est différent de id	Permet de récupérer un produit selon son nom sauf celui qui correspond à l'id fourni
countAll		count int	Permet de compter le nombre de produit
countByTypeAndSport	newtype varchar(15), newsport int	count int	Permet de compter le nombre de produits par sport et par type
getProductByRangeAndSportAndType	start int, step int, sport int, newtype varchar(15)	Renvoie les lignes qui commence de start est qui fini à step de product où le nusport est égale à sport, le type est égale à newtype et isALive qui vaut true	Permet d'avoir un nombre donné de produits par sport et par type, en commençant à partir de start et en prenant step produits
revive	newid int		Permet de faire revivre un produit donc mettre son isALive à true

Package Order

Nom	Paramètre entrant	Paramètre sortant	Commentaire
-----	-------------------	-------------------	-------------

Nom	Paramètre entrant	Paramètre sortant	Commentaire
getOrderUserById	nuorder int, newiduser int	id int, iduser int, date(dateorder) as dateorder datetime	Permet d'avoir une commande par son ID pour un client donné
getOrderUser	iduser int	id int	Permet d'avoir les commandes d'un client
addOrder	newid int, newiduser int, newdateorder datetime, newidlocation int, newstate varchar(15), newisalive bool, newdatelastupdate datetime		Permet d'ajouter une commande à un client
addProductToOrder	newidorder int, newidproduct int, newidvariant int, newquantity int		Permet d'ajouter un produit à une commande
updateState	nuorder int, newstate varchar(15)		Permet de mettre à jour l'état d'une commande
updateLocationOrder	nuorder int, newlocation varchar(15)		Permet de mettre à jour l'adresse d'une commande
getAll		Toutes les lignes de order	Permet de récupérer toutes les commandes
verifyId	newid int	id int	Permet de vérifier que l'ID n'est pas déjà présent dans la base de données
delOrder	newidorder int		Permet de supprimer une commande
getOrderProduct	id int	Toutes les lignes de order_products où l'idorder vaut id	Permet de récupérer tous les produits d'une commande

Nom	Paramètre entrant	Paramètre sortant	Commentaire
getOrderById	newid int	Toutes les lignes de order où l'id est égale à newid	Permet de récupérer une commande à partir de son ID
getAllProduct		Toutes les lignes de order_products	Permet de récupérer tous les produits de toutes les commandes
getAllWithInfo		id int, dateorder date, state varchar, iduser int, idlocation int, name varchar, firstname varchar	Permet de récupérer toutes les commandes avec les informations de l'utilisateur

Package Catalog

Nom	Paramètre entrant	Paramètre sortant	Commentaire
getStock	id integer	Toutes les lignes de catalog avec le nuproduit est égale à id et isALive qui vaut true	Permet d'avoir le stock d'un produit par son ID, donc toutes les variantes existantes
getStockTotal	id integer	total as sum(quantity) int	Permet d'avoir le nombre total en stock d'un produit
getStockByVariant	newidvariant int	quantity int	Permet d'avoir le stock d'une variante d'un produit
addCatalog	newid int, newproduit int, newreference long, newcouleur varchar(20), newtaille varchar(3), newquantite int, newIsALive bool, newDate datetime		Permet d'ajouter au Catalogue un nouveau produit ou variante d'un produit
delVariante	idvariante int		Permet de supprimer une variante

Nom	Paramètre entrant	Paramètre sortant	Commentaire
updateQuantity	idvariante int, newquantity int		Permet de mettre à jour la quantité d'une variante donnée
updateALive	idvariante int, newstate bool		Permet de mettre un variant de produit alive avec son id
getAll		Toutes les lignes de catalog	Permet de récupérer toutes les variantes de produits
getCatalogByVariant	newidvariant int	Toutes les lignes de catalog où l'id est égale au newidvariant	Permet de récupérer une variante à partir de son catalogue
getAllByNuProduct	newNuProduct int	Toutes les lignes de catalog où le nuproduct est égale à newNuProduct	Permet de récupérer toutes les variantes d'un produit à partir de son numéro
getCatalogById	newid int	Toutes les lignes de catalog où l'id est égale à newid	Permet de récupérer une variante à partir de son catalogue
updateCatalogQuantite	newid int, newquantite int		Permet de mettre à jour la quantité d'une variante dans le catalogue
deleteCatalog	newid int		Permet de supprimer une variante dans le catalogue
heHaveCatalog	newnuproduct int, newColor varchar(20), newSize varchar(3)	count int	Permet de savoir si une variante existe avec se numéro, cette couleur et cette taille

Nom	Paramètre entrant	Paramètre sortant	Commentaire
getStockAll	id integer	Toutes les lignes de catalog où l'id est égale à nuproduit et isALive vaut true	Permet d'avoir tout le stock

Package Captcha

Nom	Paramètre entrant	Paramètre sortant	Commentaire
countWordCapchat	newWord varchar(20), ipAddress varchar(45), expiration int	count int	Permet d'avoir le nombre de Captcha actif pour une adresse donnée
addCaptchat	captchaTime int, ipAddress varchar(45), newWord varchar(20)		Permet d'ajouter un Captcha à la base de données
cleanCaptchat	expiration int		Permet de supprimer un Captcha en fonction de sa durée de vie

Package VerifKey

Nom	Paramètre entrant	Paramètre sortant	Commentaire
verifyId	id varchar(54)	id int	Permet de vérifier si l'ID est valide
verifyKey	newKey varchar(6)	keyValue varchar(6)	Permet de vérifier si la clé est valide
createKey	newId varchar(54), newKey varchar(6), newDateCreation datetime, newDateExpiration datetime, newIdUser int		Permet de créer une clé
allIdKey		id int	Permet de récupérer toutes les clés
deleteKey	searchId varchar(54)		Permet de supprimer une clé

Nom	Paramètre entrant	Paramètre sortant	Commentaire
checkCode	newId varchar(54), newKey varchar(6)	idUser int, dateExpiration datetime	Permet de vérifier si le code est valide
getIdByKey	newId varchar(6)	idUser int	Permet de récupérer l'ID à partir d'une clé

Package Cart

Nom	Paramètre entrant	Paramètre sortant	Commentaire
getCart		Toutes les lignes de la table Cart	Permet de récupérer tous les paniers
addCart	newid int, newidUser int, newidcart int, newidvariant int, newquantity int, newdate datetime, newdateexp datetime		Permet d'enregistrer un panier
verifyId	id int	idcart int	Permet de vérifier si l'ID n'est pas déjà présent dans la base
getCartById	newid int	Toutes les lignes de cart avec l'iduser correspondant à newid	Permet de récupérer un panier avec son ID
maxIdCart	newid int	MAX(idcart) int	Permet de récupérer l'ID du panier le plus grand d'un utilisateur
modifyQuantity	newquantity int, user int, cart int, variant int		Permet de modifier la quantité d'un panier
deleteCart	newidcart int, newidUser int		Permet de supprimer un panier

Nom	Paramètre entrant	Paramètre sortant	Commentaire
deleteProductDB	newiduser int, newid int		Permet de supprimer un produit d'un panier
getCartIdcart	newiduser int ,newidcart int	Toutes les lignes de cart avec l'iduser correspondant à newiduser et l'idcart égale à newidcart	Permet de récupérer un panier avec son ID