



RAPPORT DE PROJET UE PROJET BD/RESEAU L3 I

Station de casier connectés

ZHANG Qinyu

Alex VOLQUARSEN

Matthieu VAYSSE

Version du rapport : 1.4

Septembre 2022 – Mai 2022

Responsable de TD : Marc Lemaire

Licence MIPI parcours informatique L3

Sommaire

Table des matières

I.	Description du projet :	3
1)	Contexte du projet	3
2)	Fonctionnement	3
II.	Dictionnaire des données.....	4
III.	Schéma des échanges réseau.....	6
IV.	Schéma E/A.....	7
V.	Schéma relationnel.....	8
1)	Cas n°1 : Fermeture d'un casier :	8
2)	Cas n°2 : Ouverture d'un casier :	9
3)	Cas n°3 : L'ouverture n'est pas possible :	10
4)	Cas n°4 : La fermeture est impossible :	11
VI.	Diagramme applicatif	12
VII.	Jeu de données :	12
1)	Les différents utilisateurs :	12
2)	Les différents étudiants :	13
3)	Echantillons d'enseignants :	13
4)	Echantillons de départements :	13
5)	Différents exemples de cartes d'utilisateurs :	14
6)	Exemples de périodes de cours :	14
7)	Echantillon de la table « est_dans » :	15
VIII.	Les différentes requêtes SQL utiliser par notre serveur :	16
1)	Les requêtes sur l'utilisateur :	16
2)	Les requêtes sur les casiers :	17

I. Description du projet :

1) Contexte du projet

Dans le cadre de l'UE projet BD/Réseau visant à mettre en pratique les connaissances acquises dans la matière base de données et ainsi que la matière réseau, nous avons eu l'idée d'un projet visant à créer une station de casier connecté. Ces casiers ne seront pas attirés à un ou plusieurs utilisateurs mais seront réservables en libre-service au sein de l'université. Ces casiers pourront être réservé à condition d'être un membre de l'université et d'avoir une carte étudiante ou enseignante à l'aide d'un terminal associé à chaque casier. Une fois qu'un casier est occupé par un utilisateur, seul ce même utilisateur peut débloquent le casier et il en va de même pour un casier réservé. L'utilisateur aura la possibilité à l'aide d'un site internet de, consulter la carte des casiers (pour voir les quels sont libres ou non), réserver un casier, voir les informations à propos d'une éventuelle utilisation en cours (numéro du casier, département du casier, date limite de réservation...) en se connectant à son compte.

2) Fonctionnement

Pour qu'un utilisateur puisse réserver un casier, il devra simplement passe sa carte devant le lecteur qui enverra les informations du casier ainsi que celles de l'utilisateur contenues dans la carte au serveur. Une fois la connexion au serveur et la récupération des données réussies, en fonction du nombre et du type des données, le serveur va effectuer différentes vérification (validité de la carte, quel utilisateur, casier libre...) et va retourner un message de vérification réussie si aucun problème n'a été détecté ou alors un message d'erreur sera retourné dans le cas contraire. Par la suite, au travers des autres échanges, après avoir reçues les données envoyées par le client, le serveur va se connecter à la base de données afin de récupérer ou modifier certaines de ses informations.

Par exemple, imaginons que l'étudiant X souhaite déposer son sac dans un casier, il va pouvoir voir sur le site où directement sur place lesquels sont encore disponible. Une fois sur place et après avoir trouvé un casier pour mettre son sac, il pourra poser ses affaires et fermer le casier en passant sa carte devant le lecteur de carte. Alors une lumière verte indique que le casier est bien verrouillé ou dans le cas où le casier ne peut pas être réservé, une lumière rouge sera allumée. Pour récupérer ses affaires il n'aura juste qu'à passer sa carte sur le lecteur du casier qui le débloquent. Les casiers auront des conditions d'occupation, par exemple un étudiant en informatique ne pourra accéder qu'aux casiers attribuer à son département et aura une limite de temps d'occupation et d'utilisation de casier variant selon différents critères. Les casiers seront communs aux étudiants et aux membres du personnel mais suivront des conditions différentes en fonction du statut de l'utilisateur. Les casiers seront réservables par avance sur le site, encore une fois selon des conditions pour ne pas permettre à quelqu'un de bloquer un casier réservé qu'il n'utilisera pas.

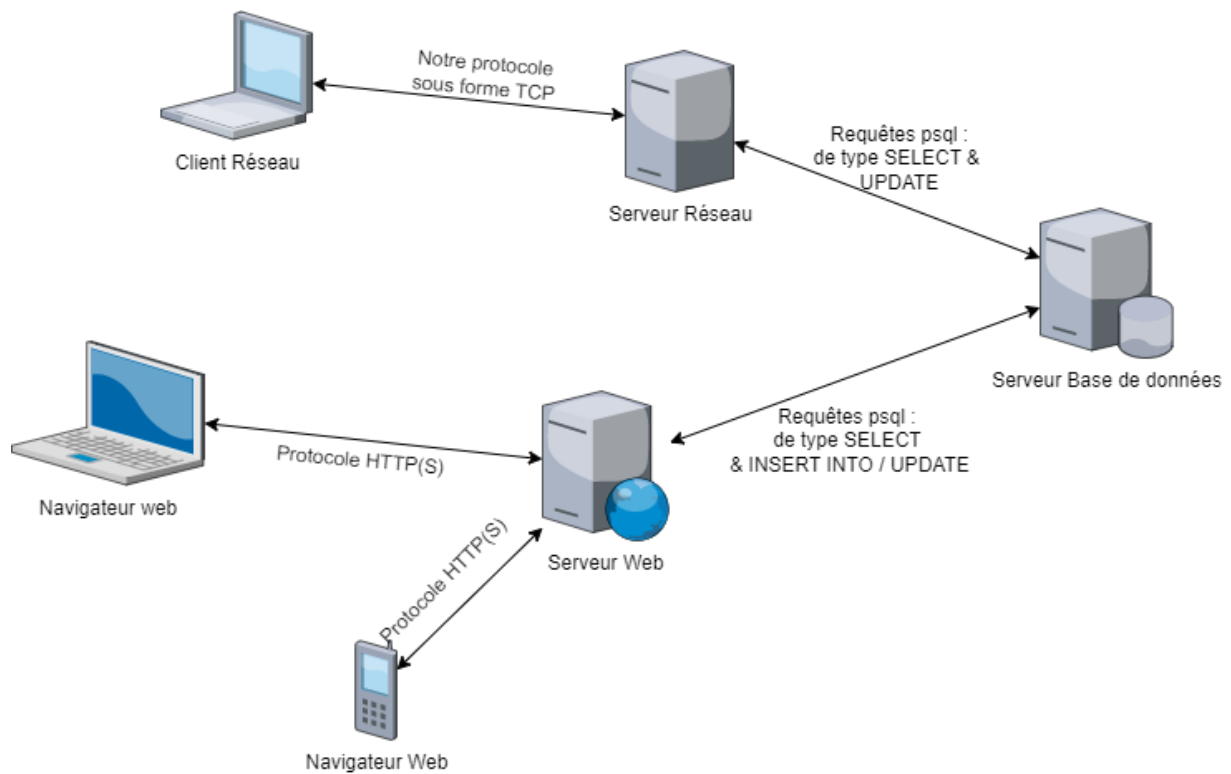
II. Dictionnaire des données

Attribut	Type	Domaine	Null ou N.N	Exemple
Prenom	varchar(25)	alphabétique	N.N	« Romain » / « Jean-Eude » / ...
Nom	varchar(25)	Alphabétique En majuscule	N.N	« GROSJEAN » / « ALLAINGUILLAUME »
Utilisation_en_cours	Int	Numérique ≥ 0	Null	Le nombre de casier utilisé par l'utilisateur ; par exemple : « 0 » / « 1 » / ...
Penalty	Bool	Booléen	N.N	Définit si l'utilisateur a une pénalité en cours
Penalty_time	Time	Strictement > 0	Null	Temps de pénalité restant à l'utilisateur ; par exemple : « 24 :00 :00 » / « 00 :30 :00 » /...
Emplois_du_temps	Time	Compris entre 08 :00 :00 & 18 :00 :00	N.N	Heure de début et de fin d'une journée
Id_département	Varchar(30)	alphabétique	N.N	« informatique » / « physique » / ...
Birthday	DATE	Compris entre 01\01\1950 & 01\01\2005	N.N	« 30/02/2002 » / ...
téléphone	int	numérique len() == 10 commence par 06 ou 07	Null	« 0555434050 » / « 0422521010 » /...
handicap	bool	Booléen	N.N	L'utilisateur est en situations de handicap
Mail	VarChar(50)	Alphanumérique (contient « @cyu.fr »)	N.N	« gerard.menvussa@cyu.fr » / « godo95001@cyu.fr » / ...
id_user	Char(10)	alphanumérique	N.N	« 22 100 493 » / « 22 100 517 »/...
Groupe_TD	Char(8)	alphabétique	N.N	« groupe D » / « groupe A » / ...
Groupe_CM	Char(12)	alphanumérique	N.N	« Groupe CM S5 » / ...
Filiere	Varchar(20)	alphabétique	N.N	« physique » / « CUPGE » /...
Tuteur	Booléen	Bool	N.N	L'étudiant est un tuteur ou non
Année_étude	Char(9)	alphanumérique	N.N	« 3 ^{ème} année » / « 1 ^{er} année » / ...

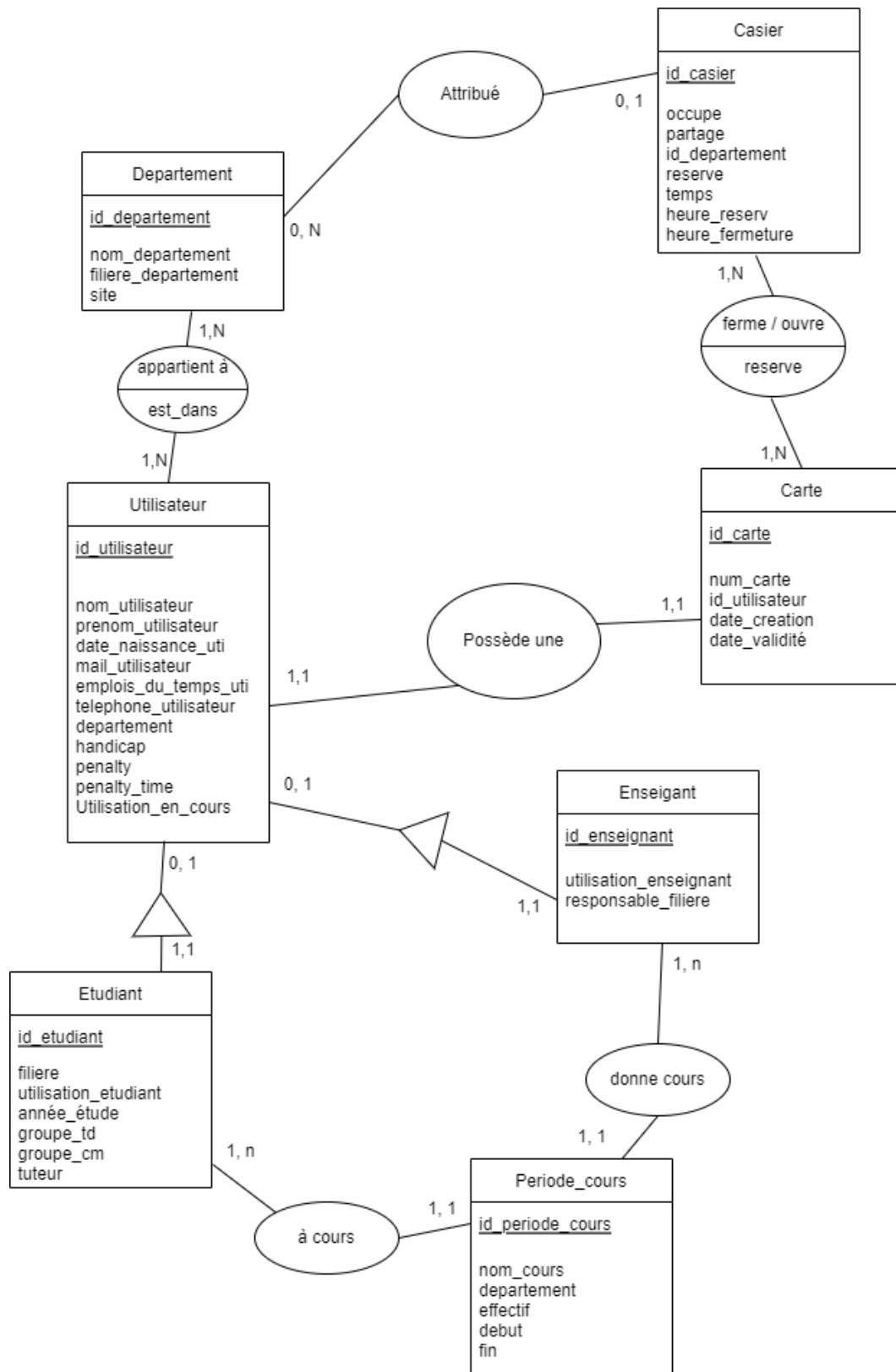
Attribut	Type	Domaine	Null ou N.N	Exemple
Utilisation_etu	Int	Numérique >= 0	Null	Le nombre de casier utilisé par l'étudiant
id_cours	Varchar(25)	alphabétique	N.N	«Base de données » / ...
Effectif_periode	int	Numérique >= 0	NN	Le nombre d'étudiant maximum présent lors de la période de cours
Début	Time	Compris entre 08 :00 :00 & 18 :00 :00	N.N	Heure du début du cours ; par exemple : « 12 :45 » / « 08 :30 » / ...
Id_enseignant	Char(10)	alphanumérique	N.N	« 22 100 493 » / « 22 100 517 »/...
Utilisation_prof	Int	Numérique >= 0	Null	Le nombre de casier utilisé par le professeur
id_locker	Char(9)	Alphanumérique Contient « locker »	N.N	« locker_43 »
Occupé	bool	Booléen	N.N	Est-ce que le casier est utilisé ?
Reserve	bool	Booléen	N.N	Est-ce que le casier est réservé ?
Time	Time	>= 0	Null	« 3 :30 :00 / ...
Heure_reservation	Time	== heure actuel lors de reservation	Null	« 15 :30 » / « 08 :17 » /...
Heure_fermeture	Time	== heure actuel lors de fermeture	Null	« 13 :00 » / « 08 :15 » /...
partage	Varchar(200)	alphanumérique	N.N	Si le casier est partagé, contient le l'id de tout les utilisateurs
Id_carte	int	Numérique Len() == 8	N.N	« 22221122 » / « 33331133 »
Num_carte	Char(14)	Numérique	N.N	« 1111 2222 3333 »
Code_carte	Char(4)	Numérique	N.N	« 4973 »
Date_création	Date	Date	N.N	Date de création de la carte
Date_validité	Date	Date	N.N	Date à laquelle la carte ne sera plus valide
Id_departement	Char(4)	Numérique	N.N	« 1001 » / « 1002 »
Nom_departement	Varchar(20)	alphanumérique	N.N	« informatique » / « mathématiques »
Lieu	Varchar(30)	Alphabétique	N.N	« Saint Martin » / « Hirsch » / ...
Filières	Varchar(200)	alphabétique	N.N	« informatique,cupge »
Date_appartenance	Date	« dmy » Forcément au plus 1 mois après début de l'année scolaire	N.N	« 01-09-2023 »

Attribut	Type	Domaine	Null ou N.N	Exemple
Bloqué	Bool	Booléen	N.N	Bloqué : « True » / « False »

III. Schéma des échanges réseau



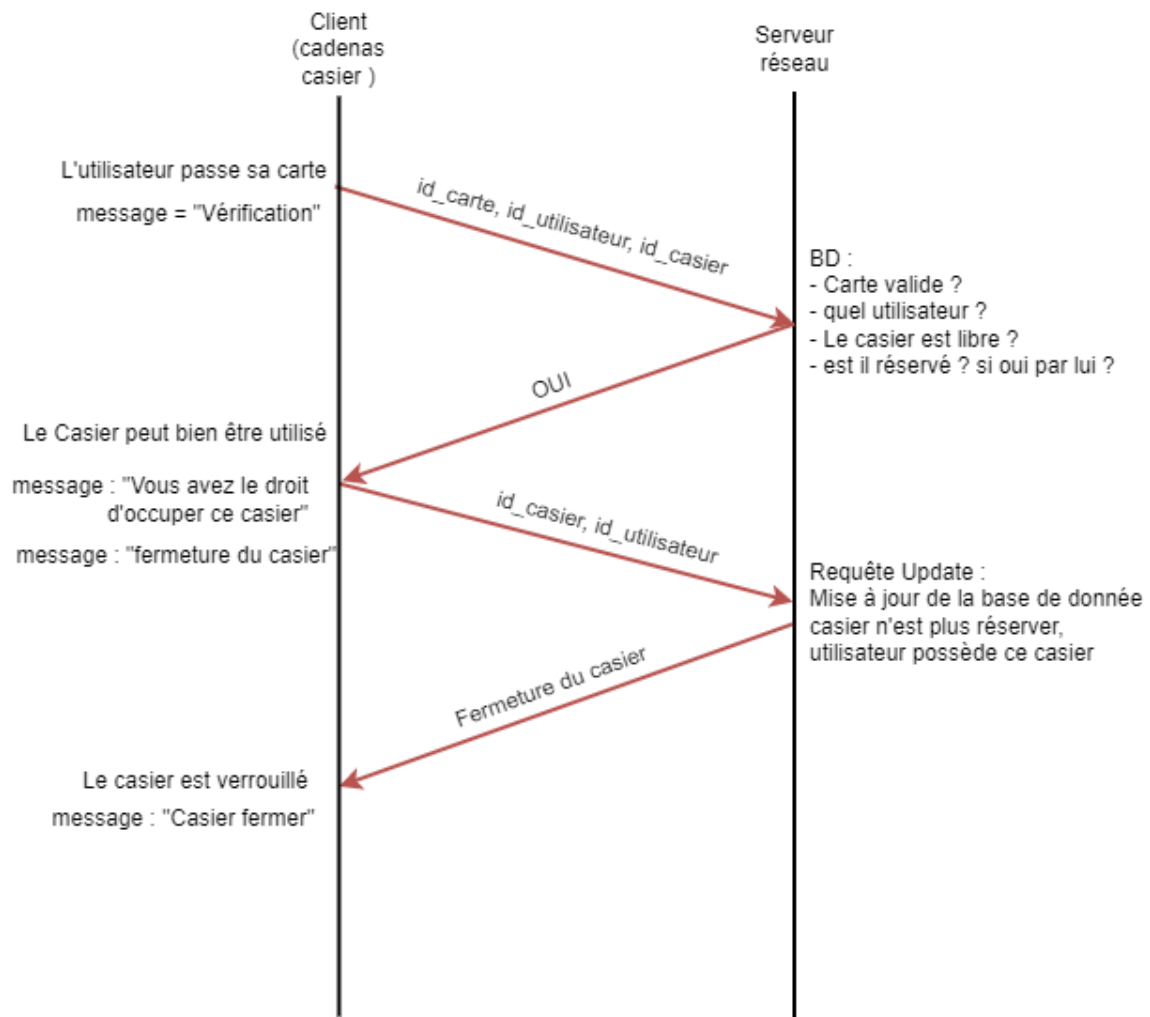
IV. Schéma E/A



V. Schéma relationnel

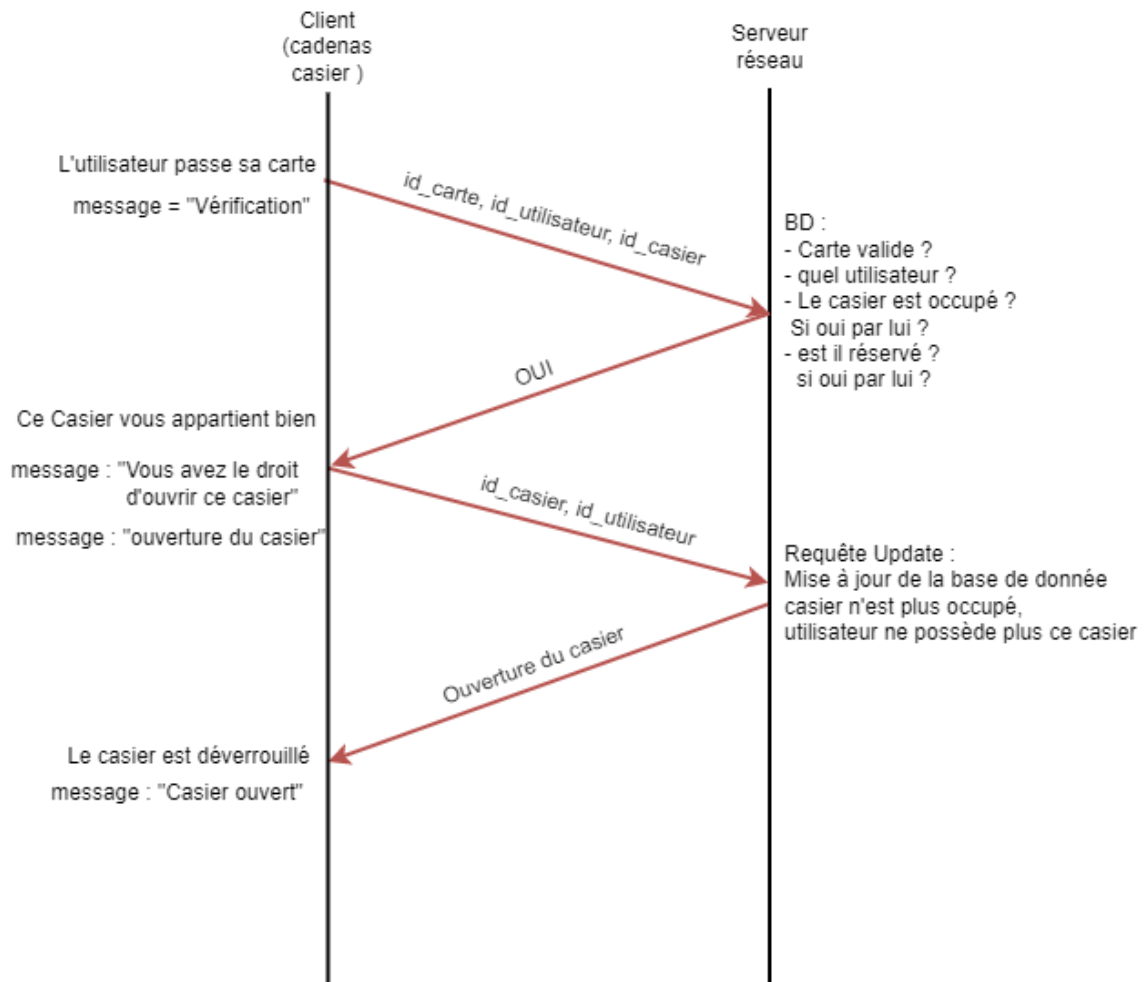
1) Cas n°1 : Fermeture d'un casier :

Dans ce cas, l'utilisateur en passant sa carte sur le client va lui permettre d'envoyer des informations au serveur réseau. Le serveur réseau va par la suite vérifier que les informations sont correctes et si c'est le cas, alors il envoie une confirmation au client, qui notifiera à l'utilisateur que le casier va être fermé puis le serveur va envoyer une requête UPDATE à la base de données pour modifier l'état du casier et le nombre d'utilisation de l'utilisateur ?



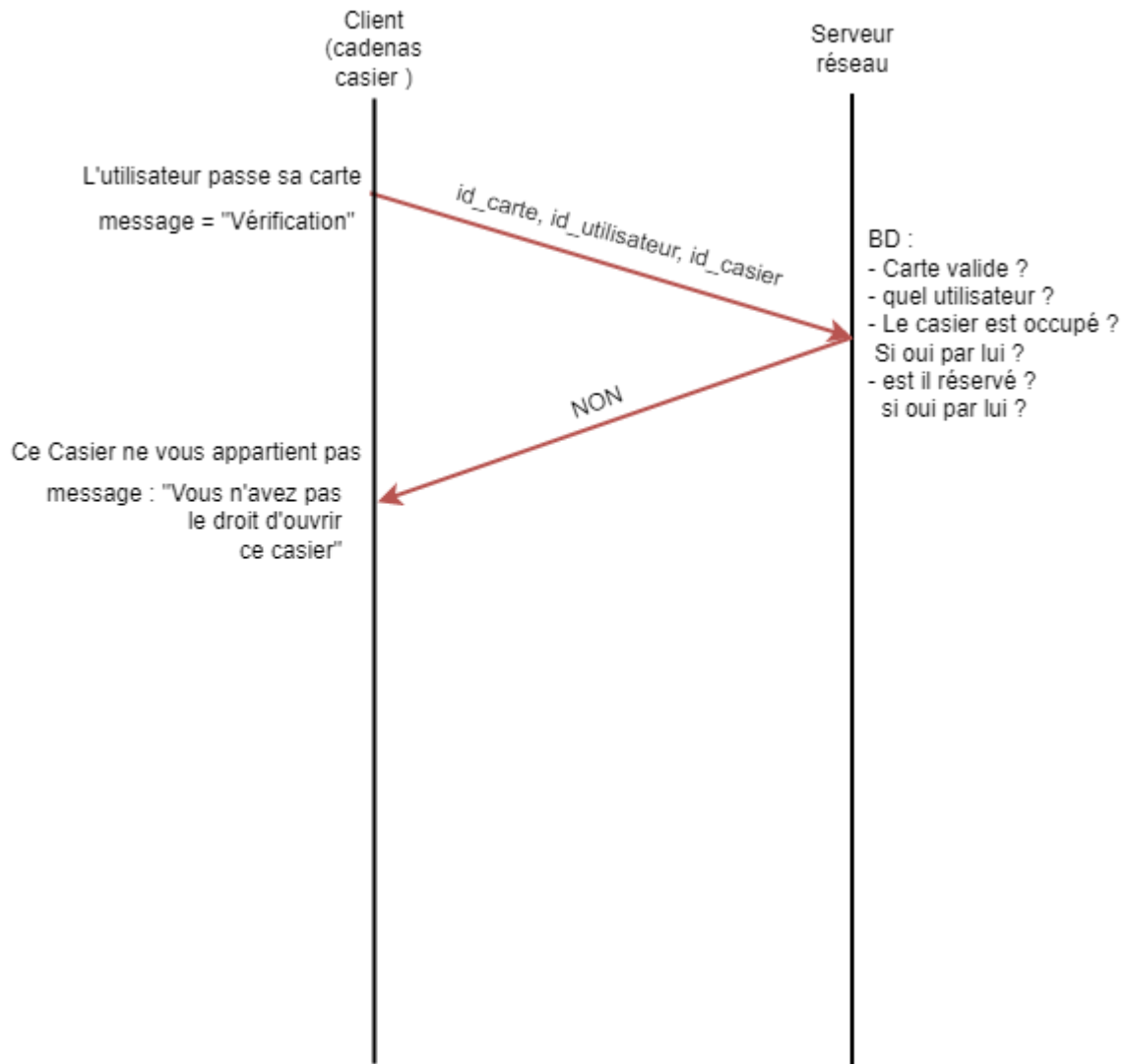
2) Cas n°2 : Ouverture d'un casier :

Dans ce cas le casier est occupé par un utilisateur, après l'envoi d'informations par le client, le serveur réseau vérifie qu'elles sont correctes, si c'est le cas, cela veut dire que le casier appartient bien à l'utilisateur et que donc il peut l'ouvrir. Le client affiche donc un message de validation et le serveur envoie des requêtes à la base de données pour rendre le casier disponible et enlève une utilisation de l'utilisateur.



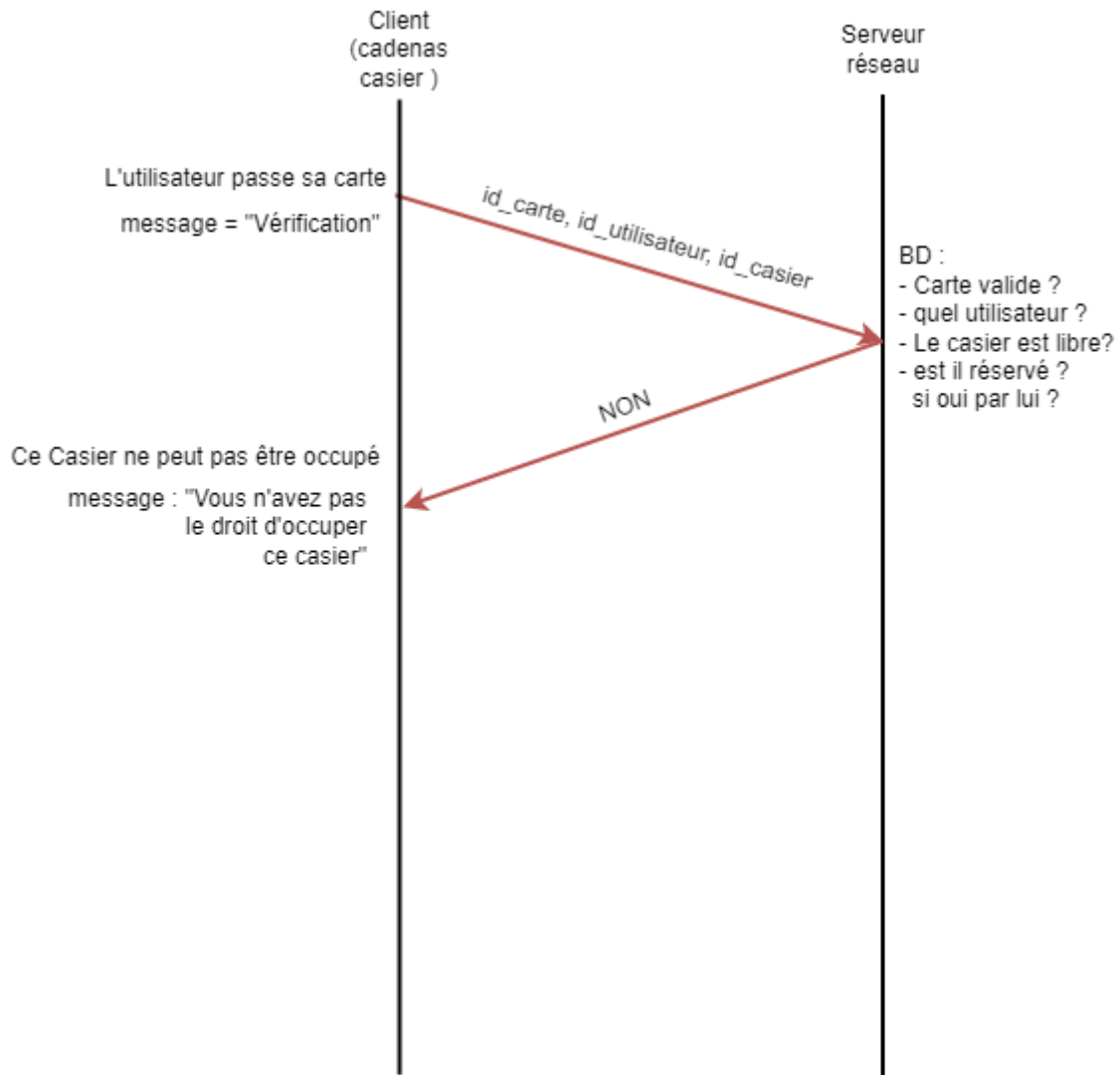
3) Cas n°3 : L'ouverture n'est pas possible :

Dans ce cas, l'utilisateur essaye d'ouvrir un casier mais après vérification par le serveur, il n'a pas réservé ni ne possède le casier donc le client lui affiche un message d'erreur et aucune action n'est effectuée sur la base de données.



4) Cas n°4 : La fermeture est impossible :

Lorsque l'utilisateur essaye de fermer le casier, si ce dernier est déjà réservé ou occupé, ou que ses informations sont incorrectes, alors le client lui affiche un message d'erreur et aucune action n'est effectuée sur la base de données.



VI. Diagramme applicatif

Casier (id_casier, occupe, partage, departement, reserve, temps, heure_reservation, heure_fermeture, #id_departement, #id_utilisateur)

Carte (id_carte, date_creation, date_validite, #id_utilisateur)

Reserve (#id_casier, #id_carte)

Departement (id_departement, nom_departement, lieu, filiere)

Utilisateur (id_utilisateur, nom, prenom, birthday, email, telephone, utilisation_en_cours, penalite, temps_penalite, handicap, #departement)

Est_dans (#id_utilisateur, #id_departement)

Etudiant (#id_etudiant, annee_etude, filiere, groupe_cm, groupe_td, utilisation_etu, tuteur)

Enseignant (#id_enseignant, utilisation_enseignant, responsable_filiere).

Periode_cours (id_cours, nom, effectif, debut, fin, #departement)

VII. Jeu de données :

Dans notre base de données, nous allons retrouver des échantillons de données pouvant servir d'exemple lors de l'exécution des différentes fonctions de notre projet.

1) Les différents utilisateurs :

Nous pouvons voir ci-dessous un exemple de différents utilisateurs ayant des attributs différents.

Actions	id_utilisateur	nom	prenom	birthday	mail	telephone	departement	utilisation_en_cours	penalty	penalty_time	handicap
Éditer	Effacer	22 100 493	VAYSSE	Maththieu	2002-07-05	matthieu.vaysse2002@cyu.fr	643316947 ☎=1001	1	FALSE	NULL	FALSE
Éditer	Effacer	22 100 517	VOLQUARSEN	Alex	2001-12-16	alex.vloquardsen@cyu.fr	679801313 ☎=1001	1	FALSE	NULL	FALSE
Éditer	Effacer	22 100 400	LEMAIRE	Marc	1968-10-24	marc.lemaire@cyu.fr	679561327 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 401	LEMATEU	Gerard	1979-10-13	gerard.lemateu@cyu.fr	647693821 ☎=1002	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 402	OBUCHET	Jeanne	1982-01-01	jeanne.obuchet@cyu.fr	614312000 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 403	LAGROSTET	David	1980-05-20	gerard.lemateu@cyu.fr	647693821 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 415	VITO	WAYANE	1990-11-22	WAYANE.VITO@cyu.fr	677790327 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 114	REMY MORIN	GUYVELINE	1987-08-10	GUYVELINE.REMY.MORIN@cyu.fr	657808624 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 333	HLIHAL	DJAWED	2003-11-23	DJAWED.HLIHAL@cyu.fr	682496524 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 087	CAPDOROY	SCHNEIDER	1974-09-08	SCHNEIDER.CAPDOROY@cyu.fr	600440880 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 921	ROCHE FABRE	MARIA-MERCEDES	1966-06-08	MARIA-MERCEDES.ROCHE.FABRE@cyu.fr	674506115 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 804	AMGUINE	GUILIANE	1993-09-22	GUILIANE.AMGUINE@cyu.fr	685781129 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 459	ROSE FELDMAN SEDGWICK	INTISAR	1994-03-03	INTISAR.ROSE.FELDMAN.SEDGWICK@cyu.fr	640816620 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 993	EZZAOUINI	ABDOULAH	1995-10-16	ABDOULAH.EZZAOUINI@cyu.fr	639223346 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 630	MACIUGA	CHAHINE	1970-08-22	CHAHINE.MACIUGA@cyu.fr	617398801 ☎=1002	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 829	NKOY OKITEMBO	HAWA	1965-04-17	HAWA.NKOY.OKITEMBO@cyu.fr	648376163 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 707	FRISSET	KARMEN	1955-05-02	KARMEN.FRISSET@cyu.fr	636144791 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 820	LAATEUR	MELISSANDRE	1953-03-01	MELISSANDRE.LAATEUR@cyu.fr	657545273 ☎=1004	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 859	OKHISSI	YVETTE	1999-10-12	YVETTE.OKHISSI@cyu.fr	623220570 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 964	ASSOGRANI	DIMAS	1977-01-24	DIMAS.ASSOGRANI@cyu.fr	628651859 ☎=1002	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 149	GOUQUE	MARIA-SOL	1959-11-16	MARIA-SOL.GOUQUE@cyu.fr	659582830 ☎=1002	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 182	FEUSIER	LEANDRY	1984-07-13	LEANDRY.FEUSIER@cyu.fr	601188197 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 480	BARRERAS GOMEZ	GRACIEUSE	2002-04-09	GRACIEUSE.BARRERAS.GOMEZ@cyu.fr	635315761 ☎=1002	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 872	HOSDAIN	MAHENDRA	2004-01-25	MAHENDRA.HOSDAIN@cyu.fr	622704840 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 308	FONDINI	DEKLAN	1971-10-09	DEKLAN.FONDINI@cyu.fr	691642591 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 704	PINOT DE VILLECHENON	KHIERA	1993-02-22	KHIERA.PINOT.DE.VILLECHENON@cyu.fr	693584637 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 723	DUHOTY	ELYESS	2004-01-26	ELYESS.DUHOTOY@cyu.fr	634353986 ☎=1003	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 561	SEHHOUL	ACIA	2000-06-21	ACIA.SEHHOUL@cyu.fr	699555488 ☎=1001	0	FALSE	NULL	FALSE
Éditer	Effacer	22 100 347	CHELMA	DION	2004-01-09	DION.CHELMA@cyu.fr	650495376 ☎=1004	0	FALSE	NULL	FALSE

2) Les différents étudiants :

Nous retrouvons ci-dessous un échantillon de différents étudiants.

Actions		id_etudiant	annee_etude	filiere	groupe_cm	groupe_td	utilisation_etudiant	tuteur
Éditer	Effacer	🔑 22 100 493	2022-2023	L3-I	Groupe CM S5	groupe D	0	FALSE
Éditer	Effacer	🔑 22 100 517	2022-2023	L3-I	Groupe CM S5	groupe D	0	FALSE
Éditer	Effacer	🔑 22 100 415	2022-2023	L2-I	Groupe CM S3	groupe A	0	TRUE
Éditer	Effacer	🔑 22 100 114	2022-2023	L2-I	Groupe CM S3	groupe C	0	FALSE
Éditer	Effacer	🔑 22 100 087	2022-2023	L3-M	Groupe CM S5	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 630	2022-2023	L2-M	Groupe CM S3	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 820	2022-2023	L2-M	Groupe CM S3	groupe B	0	TRUE
Éditer	Effacer	🔑 22 100 872	2022-2023	L2-P	Groupe CM S3	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 308	2022-2023	L3-P	Groupe CM S5	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 561	2022-2023	L3-P	Groupe CM S5	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 347	2022-2023	L3-CUPGE	Groupe CM S5	groupe A	0	FALSE
Éditer	Effacer	🔑 22 100 723	2022-2023	L3-CUPGE	Groupe CM S5	groupe A	0	FALSE

3) Echantillons d'enseignants :

Actions		id_enseignant	utilisation_prof
Éditer	Effacer	🔑 22 100 400	0
Éditer	Effacer	🔑 22 100 401	0
Éditer	Effacer	🔑 22 100 402	0
Éditer	Effacer	🔑 22 100 403	0

4) Echantillons de départements :

Nous retrouvons dans cette image différents départements qui permettront d'identifier les étudiants et les casiers en fonction de cette table.

Actions		id_departement	nom_departement	lieu	filiere
Éditer	Effacer	1001	Informatique	Saint-Martin	L1-MIPI, L2-I, L3-I
Éditer	Effacer	1002	Math,matiques	Saint-Martin	L1-MIPI, 10000004L2-M, L3-M
Éditer	Effacer	1003	Physique	Saint-Martin	L1-MIPI, L2-P, L3-P
Éditer	Effacer	1004	CUPGE	Saint-Martin	L1-MIPI, L2-CUPGE, L3-CUPGE

5) Différents exemples de cartes d'utilisateurs :

Nous retrouverons dans notre base de données les informations des cartes de chaque utilisateur ce qui permettra au serveur d'identifier à qui appartient chaque carte. Chaque utilisateur à une unique carte.

Actions		id_carte	uti	num	code	date_creation	date_validite
Éditer	Effacer	00001030	🔑22 100 493	4470 3456 7888	2742	2020-08-31	2023-07-01
Éditer	Effacer	00001031	🔑22 100 517	4479 3416 7848	4378	2020-08-31	2023-07-01
Éditer	Effacer	00001032	🔑22 100 415	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001033	🔑22 100 114	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001034	🔑22 100 087	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001036	🔑22 100 630	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001037	🔑22 100 820	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001038	🔑22 100 872	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001039	🔑22 100 308	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001040	🔑22 100 561	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001041	🔑22 100 347	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	00001042	🔑22 100 723	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	10000001	🔑22 100 400	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	10000002	🔑22 100 401	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	10000003	🔑22 100 402	0070 3406 7988	9341	2015-08-31	2025-07-01
Éditer	Effacer	10000004	🔑22 100 403	0070 3406 7988	9341	2015-08-31	2025-07-01

6) Exemples de périodes de cours :

Dans notre base de données nous auront aussi les différentes périodes de cours afin d'identifier le besoin et les droits d'utilisation de casier par les étudiants selon leurs cours.

Actions		id_cours	nom	departement	effectif	debut	fin
Éditer	Effacer	00205	TD-3 projet BD/R,seau	🔑1001	25	13:45:00	16:15:00
Éditer	Effacer	00206	CM Algo et SDA	🔑1001	25	08:30:00	10:00:00
Éditer	Effacer	00102	CM droite	🔑1002	25	09:00:00	12:45:00
Éditer	Effacer	00201	TD droite	🔑1002	25	13:45:00	18:00:00
Éditer	Effacer	00212	CM point	🔑1003	25	08:45:00	13:00:00
Éditer	Effacer	00213	TD point	🔑1003	25	14:30:00	16:30:00
Éditer	Effacer	00250	CM dur	🔑1004	25	08:45:00	11:15:00
Éditer	Effacer	00218	TD compliqu,	🔑1004	25	13:00:00	16:00:00

7) Echantillon de la table « est_dans » :

Nous remarquons ici la table « est_dans » faisant office de relation entre les tables département et utilisateur. Elle contient tous les utilisateurs ainsi que le département auquel chacun appartient.

Actions		id_utilisateur	id_département
Éditer	Effacer	22 100 493	1001
Éditer	Effacer	22 100 517	1001
Éditer	Effacer	22 100 400	1001
Éditer	Effacer	22 100 401	1002
Éditer	Effacer	22 100 402	1003
Éditer	Effacer	22 100 403	1004
Éditer	Effacer	22 100 415	1004
Éditer	Effacer	22 100 114	1001
Éditer	Effacer	22 100 333	1003
Éditer	Effacer	22 100 087	1003
Éditer	Effacer	22 100 921	1003
Éditer	Effacer	22 100 804	1004
Éditer	Effacer	22 100 459	1004
Éditer	Effacer	22 100 993	1001
Éditer	Effacer	22 100 630	1002
Éditer	Effacer	22 100 829	1001
Éditer	Effacer	22 100 707	1004
Éditer	Effacer	22 100 820	1004
Éditer	Effacer	22 100 859	1001
Éditer	Effacer	22 100 964	1002
Éditer	Effacer	22 100 149	1002
Éditer	Effacer	22 100 182	1003
Éditer	Effacer	22 100 480	1002
Éditer	Effacer	22 100 872	1001
Éditer	Effacer	22 100 308	1001
Éditer	Effacer	22 100 704	1001
Éditer	Effacer	22 100 723	1003
Éditer	Effacer	22 100 561	1001
Éditer	Effacer	22 100 347	1004

VIII. Les différentes requêtes SQL utiliser par notre serveur :

Dans cette partie, nous allons retrouver un ensemble des différentes requêtes SQL que notre serveur va envoyer à la base de donner afin de vérifier les différentes informations qui ont été envoyés par le client.

1) Les requêtes sur l'utilisateur :

Nous allons effectuer différentes requêtes pour savoir si l'utilisateur est valide.

```
-- Chercher à l'aide de l'id de la carte si l'utilisateur est un étudiant --
SELECT e.id_etudiant
FROM etudiant as e
JOIN utilisateur as u
  ON e.id_etudiant = u.id_utilisateur
 WHERE u.id_utilisateur =
    (
      SELECT c.id_utilisateur
      FROM carte as c
      WHERE c.id_carte = '{utilisateur}'
    );
```

Premièrement nous allons vérifier si l'identifiant de la carte correspond à l'identifiant d'un étudiant, si c'est le cas, cela signifie que la personne essayant de réserver ou d'ouvrir un casier est un étudiant et donc nous pourrions effectuer les vérifications suivantes en conséquence. Cependant si le résultat de cette requête est null, cela signifie que l'utilisateur n'est pas un étudiant ou que la carte n'est pas associée à un utilisateur. Nous allons donc vérifier la première théorie.

```
-- Cherche à l'aide de l'id de la carte si l'utilisateur est un enseignant --
SELECT e.id_enseignant
FROM enseignant as e
JOIN utilisateur as u
  ON e.id_enseignant = u.id_utilisateur
 WHERE u.id_utilisateur =
    (
      SELECT c.id_utilisateur
      FROM carte as c
      WHERE c.id_carte = '{utilisateur}'
    );
```

Grace à cette requête, nous allons vérifier si l'identifiant de la carte est relié à un enseignant, si c'est effectivement le cas, alors c'est que la personne voulant utiliser un casier est un enseignant et nous partirons de ce principe pour nos vérifications. Cependant, si cette requête retourne un résultat null, cela signifie que la carte n'est associée à aucun utilisateur.


```
-- Modifie le nombre d'utilisation en cours --
```

```
UPDATE etudiant
SET utilisation_etudiant = {res[0]}
WHERE
(id_etudiant LIKE '{utilisateur}');
```

Cette requête permet de modifier le nombre d'utilisation en cours pour un étudiant, si ce dernier occupe un casier la valeur montras, et à contrario si l'utilisateur ouvre sont casier cette valeur baisseras.

```
-- Modifie le nombre d'utilisation en cours --
```

```
UPDATE enseignant
SET utilisation_prof = {res[0]}
WHERE
(id_enseignant LIKE '{utilisateur}');
```

Sur le même principe que la requête précédente, c'elle si va modifier le nombre d'utilisation en cours de l'enseignant donné.

```
-- Récupere le nombre d'utilisation d'un étudiant --
```

```
SELECT utilisation_etudiant
FROM etudiant
WHERE
(id_etudiant LIKE '{uti}');
```

Cette requête va nous permettre de vérifier le nombre d'utilisation en cours pour un étudiant donné. Par le même principe, nous allons utiliser la requête ci-dessous pour vérifier le nombre d'utilisation d'un enseignant.

```
-- Récupere le nombre d'utilisation d'un enseignant --
```

```
SELECT utilisation_enseignant
FROM enseignant
WHERE
(id_enseignant LIKE '{uti}');
```

2) Les requêtes sur les casiers :

Nous allons maintenant aborder les requêtes sur les casiers, Tout d'abord nous avons la requête qui va vérifier si l'utilisateur et le casier font partie du même département.

```
-- Vérifie si l'utilisateur et le casier sont du même département --
```

```
SELECT c.id_casier
FROM casier AS c
JOIN utilisateur as u
ON c.id_dep IS NULL OR c.id_dep = u.departement
WHERE
u.id_utilisateur = '22 100 493'
AND
```

```
c.id_casier = 'locker_01';
```

Si jamais cette requête retourne 'null' cela signifie que l'utilisateur et le casier ne font pas partie du même département et de ce fait l'utilisateur ne pourra pas réserver ou utiliser le casier. La requête vérifie aussi si le casier est désigné comme appartenant à aucun département.

```
-- Remet Le casier en état 'libre' --
```

```
UPDATE casier
```

```
SET
```

```
    occupe = FALSE,
```

```
    reserve = FALSE,
```

```
    heure_restant = NULL,
```

```
    heure_reservation = NULL,
```

```
    heure_fermeture = NULL,
```

```
    id_utilisateur = NULL
```

```
WHERE id_casier LIKE {locker};
```

Grace à cette requête, nous allons changer l'état d'un casier dans la base de données nous allons remettre tout ses attributs comme ils étaient lorsqu'il était inoccupé.

```
-- Ferme Le casier --
```

```
UPDATE casier
```

```
SET
```

```
    occupe = TRUE,
```

```
    id_utilisateur = '{utilisateur}'
```

```
WHERE
```

```
    (id_casier LIKE '{locker}');
```

Cette requête nous permettra de changer l'état du casier et de lui donner le statut « fermer ».

```
-- Vérifie si Le casier est occupé ou réservé --
```

```
SELECT id_casier
```

```
FROM casier
```

```
WHERE
```

```
    (id_casier LIKE '{locker}'
```

```
    AND
```

```
    occupe = FALSE
```

```
    AND
```

```
    reserve = FALSE);
```

A l'aide de cette requête, nous allons vérifier si le casier est déjà occupé ou réservé, si ce n'est pas le cas alors nous pourrions agir en conséquence sur ce dernier.