



2021

Projet Rapport



MESRAR HAMZA.
Élève en Première année Cycle Ingénieur
en informatique et ingénierie de données
à ENSA KOURIBGA



Projet du système d'informations et
bases de données sous l'encadrement
du professeur GHERABI Noredine



+212 633-099608



mesrarhamza48@gmail.com



Table de contenu :

- Introduction
- Description du Projet
- Système d'information
 - Les entités du système
 - Dictionnaire de données
 - Modèle Conceptuel du Système MCD
 - Modèle relationnel et MLD
 - Modèle conceptuel de traitement MCT
 - Modèle organisationnel de traitement MOT
- Base de Données
 - Les requêtes SQL en langage LCD
 - ▶ Créer des utilisateurs / rôles
 - ▶ Affecter des droits d'accès
 - ▶ Retirer les droits d'accès
 - Le schéma relationnel en langage LDD
 - ▶ Création du Tableaux du Système
 - Les requêtes SQL en langage LMD
 - ▶ Sélection
 - ▶ Suppression
 - ▶ Modification
 - ▶ Sélection en jointure
- Conclusion





Introduction :

La gestion des salles est devenue une compétence reconnue dans les écoles et les universités qui œuvre tant en la facilité de l'accès et réserver des salles. Dans ce projet en vont tracer une feuille de route pour la mise en place d'une de gestion des salles et l'organisation des évènements de l'Ecole Nationale des Sciences Appliquées de Khouribga. Puis qu'on n'a pas un système qui fait cette opération, C'est l'opportunité de créer un. Pour mettre n'importe quelle personne qui appartient à l'école plus familière avec les opérations comme réservation des salles, voir le planning des réservations, organisation des événements etc...

Description du Projet :

Le projet est principalement sur la gestion des salles et l'organisation des événements dans ENSA KHOURIBGA.

→ Gestion des Salles :

- # Réserver une salle
- # Voir et gérer le planning des réservations
- # Gérer les équipements des salles
- # ...

→ L'organisation d'événements :

- # Réaliser un événement
- # Voir et gérer le planning d'événements
- # ...

→ Créé les espaces et l'utilisateur :

- # Espace et utilisateurs de la direction
- # Espace et utilisateurs des professeurs
- # Espace et utilisateurs des Etudiants





Système d'information :

Le système d'information (SI) est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information.

➤ Les entités du système :

Une entité ou individu est la représentation dans le SI d'un objet concret ou abstrait (matériel ou immatériel) de l'univers de l'entreprise.

- * ETUDIANT
- * PROFESSEUR
- * ADMINISTRATEUR
- * RESERVATIONS
- * SALLES
- * BLOC
- * EQUIPEMENT
- * EVENEMENTS

➤ Dictionnaire de données :

La liste de toutes les propriétés est définie dans un dictionnaire de données. Celui-ci contient pour chaque propriété tout ou partie des éléments suivants : son nom, son format et sa longueur, sa nature, son type et contrainte...





Nom de donnée	Format	Longueur	Contraint	Type		Règle de calcul	Règle de gestion	Document
				élémentaire	calculer			
Id_Etudiant	Numérique	---	Clé Primaire	x		---		projet
Nom	Alphabétique	20	NOT NULL	x		---		projet
Prénom	Alphabétique	30	NOT NULL	x		---		projet
Age	Numérique	---	NOT NULL	x		---		projet
Ville	Alphanumérique	30	---	x		---		projet
Date naissance	Date	---	NOT NULL	x		---		projet
Filière		20	NOT NULL	x		---		projet

Id_Prof	Numérique	---	Clé Primaire	x		---		projet
Nom	Alphabétique	20	NOT NULL	x		---		projet
Prénom	Alphabétique	30	NOT NULL	x		---		projet
Département	Alphabétique	20	NOT NULL	x		---		projet

Id_admin	Numérique	---	Clé Primaire	x		---		projet
Nom	Alphabétique	20	NOT NULL	x		---		projet
Prénom	Alphabétique	30	NOT NULL	x		---		projet
Fonction	Alphabétique	20	NOT NULL	x		---		projet

Num_reservation	Numérique	---	Clé Primaire	x		---		projet
Date_réservation	Date	---	NOT NULL	x		---		projet

Num_salle	Alphanumérique	10	NOT NULL	x		---		projet
Capacité	Numérique	---	NOT NULL	x		---		projet

Code_Bloc	Alphanumérique	10	Clé Primaire	x		---		projet
Département	Alphabétique	20	NOT NULL	x		---		projet

Num_serie	Numérique	---	Clé Primaire	x		---		projet
Type	Alphabétique	20	NOT NULL	x		---		projet
Fabricant	Alphanumérique	20	NOT NULL	x		---		projet





Nombre	Numérique	---	NOT NULL	x		---		projet

Id_event	Numérique	---	Clé Primaire	x		---		projet
Sujet	Alphanumérique	50	NOT NULL	x		---		projet
Date_réalisation	Date	---	NOT NULL	x		---		projet

NB : Dans la colonne du contrainte on écrit juste clés primaire mais parfois cette Propriété est un clé étrangère à une autre entité.

➤ **Modèle Conceptuel du Système MCD :**

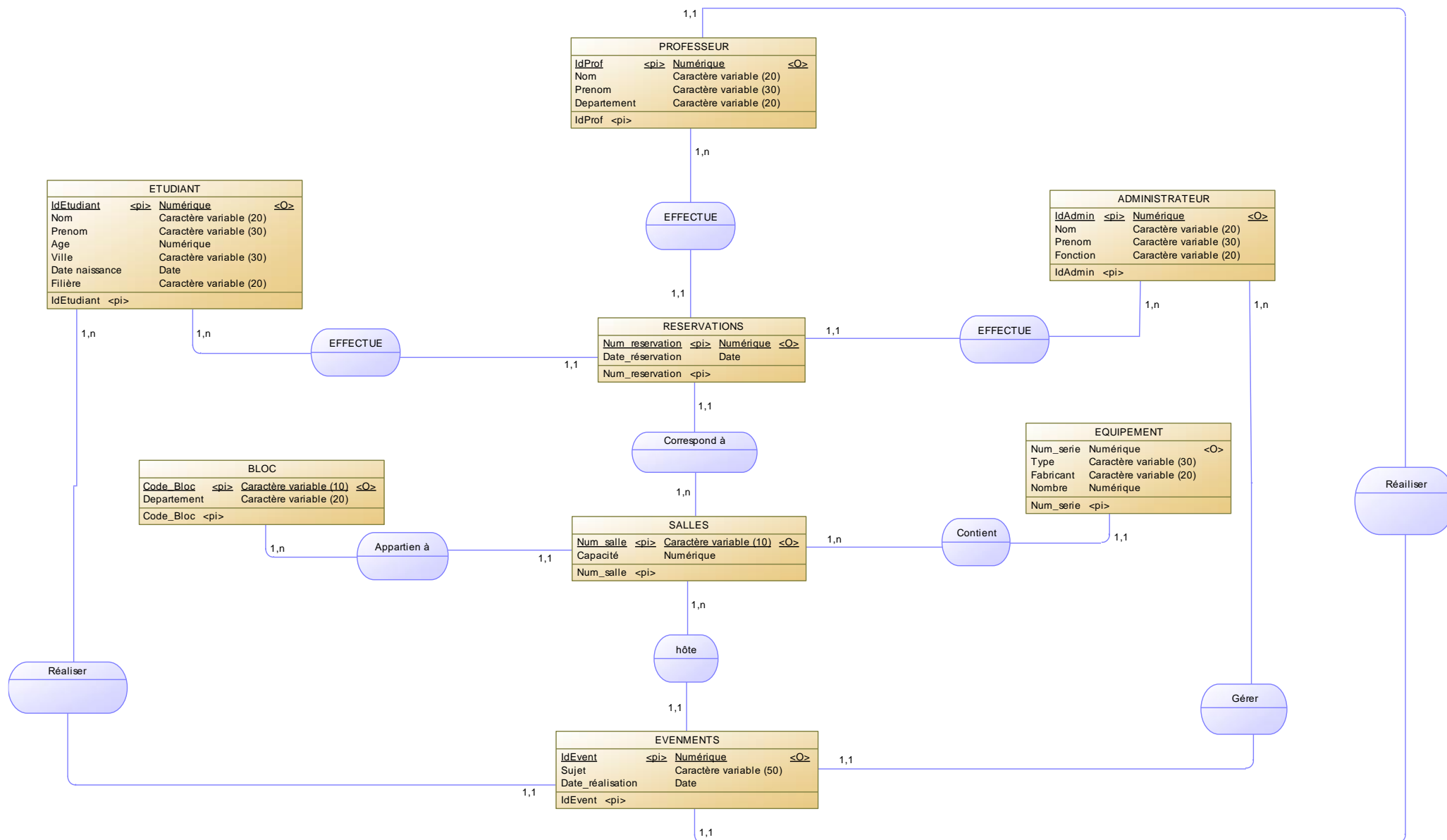
Le modèle conceptuel de données (MCD) est une représentation stable de l'ensemble des données manipulées par l'entreprise ainsi que des relations entre ces données.

▪ **Règle de gestion :**

- * Un étudiant a le droit de quatre réservations dans chaque semestre.
- * Un professeur peut réserver n'importe quelle salle au cours de l'année
- * Le système peut afficher la liste des réservations par un intervalle de date et l'heure
- * Une salle peut contenir plusieurs équipements comme vidéo projecteur, nombre de chaises, nombre de table et autres...
- * Un administrateur peut effectuer plusieurs réservations
- * L'administrateur est le seul qui peut gérer un événement.

▪ **Modèle conceptuel de données MCD :**







➤ Modèle relationnel et MLD :

▪ *Modèle relationnel du système :*

ETUDIANT (*IdEtudiant*, Nom, Prenom, Age, Ville, DateNaissance, Filiere)

PROFESSEUR (*IdProf*, Nom, Prenom, Departement)

ADMINISTRATEUR (*IdAdmin*, Nom, Prenom, Fonction)

Bloc (*CodeBloc*, Departement)

SALLES (*NumSalle*, Capacite, *#CodeBloc*)

RESERVATIONS (*NumReservation*, DateReservation, *#IdProf*, *#IdEtudiant*, *#IdAdmin*, *#NumSalle*)

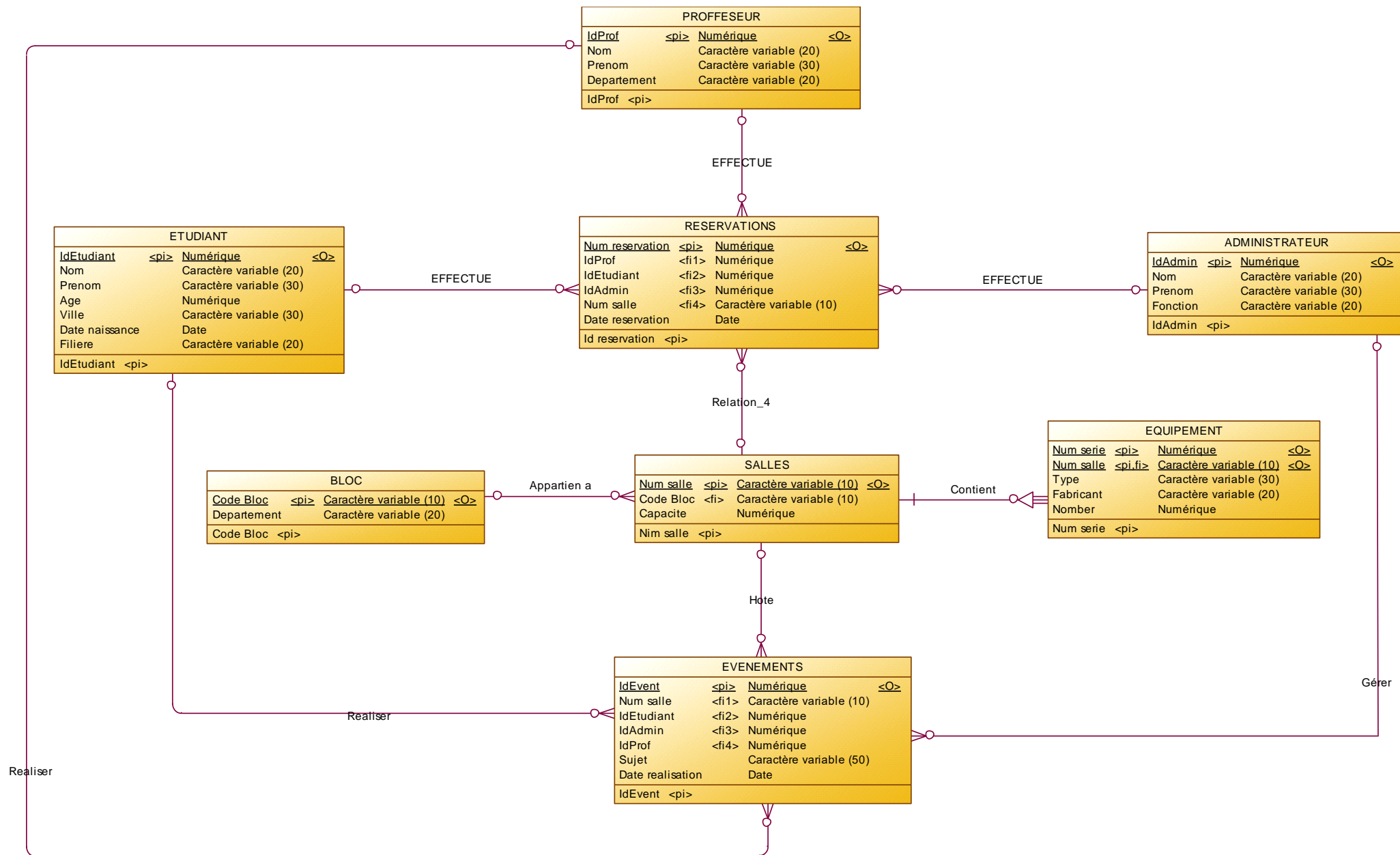
EQUIPEMENT (*NumSerie*, *#NumSalle*, Type, Fabricant, Number)

EVENEMENTS (*IdEvent*, Sujet, DateRealisation, *#NumSalle*, *#IdEtudiant*, *#IdProf*, *#IdAdmin*)

▪ *Modèle Logique de données MLD :*

Un modèle logique de données (MLD) est la représentation des données d'un système d'information. Les données sont représentées en prenant en compte le modèle technologique qui sera utilisée pour leur gestion.







➤ **Modèle conceptuel de traitement MCT :**

Le modèle conceptuel des traitements permet de traiter la dynamique du système d'information, c'est-à-dire les opérations qui sont réalisées en fonction d'événements. Ce modèle permet donc de représenter de façon schématique l'activité d'un système d'information sans faire référence à des choix organisationnels ou des moyens d'exécution, c'est-à-dire qu'il permet de définir simplement ce qui doit être fait, mais il ne dit pas quand, comment ni où...

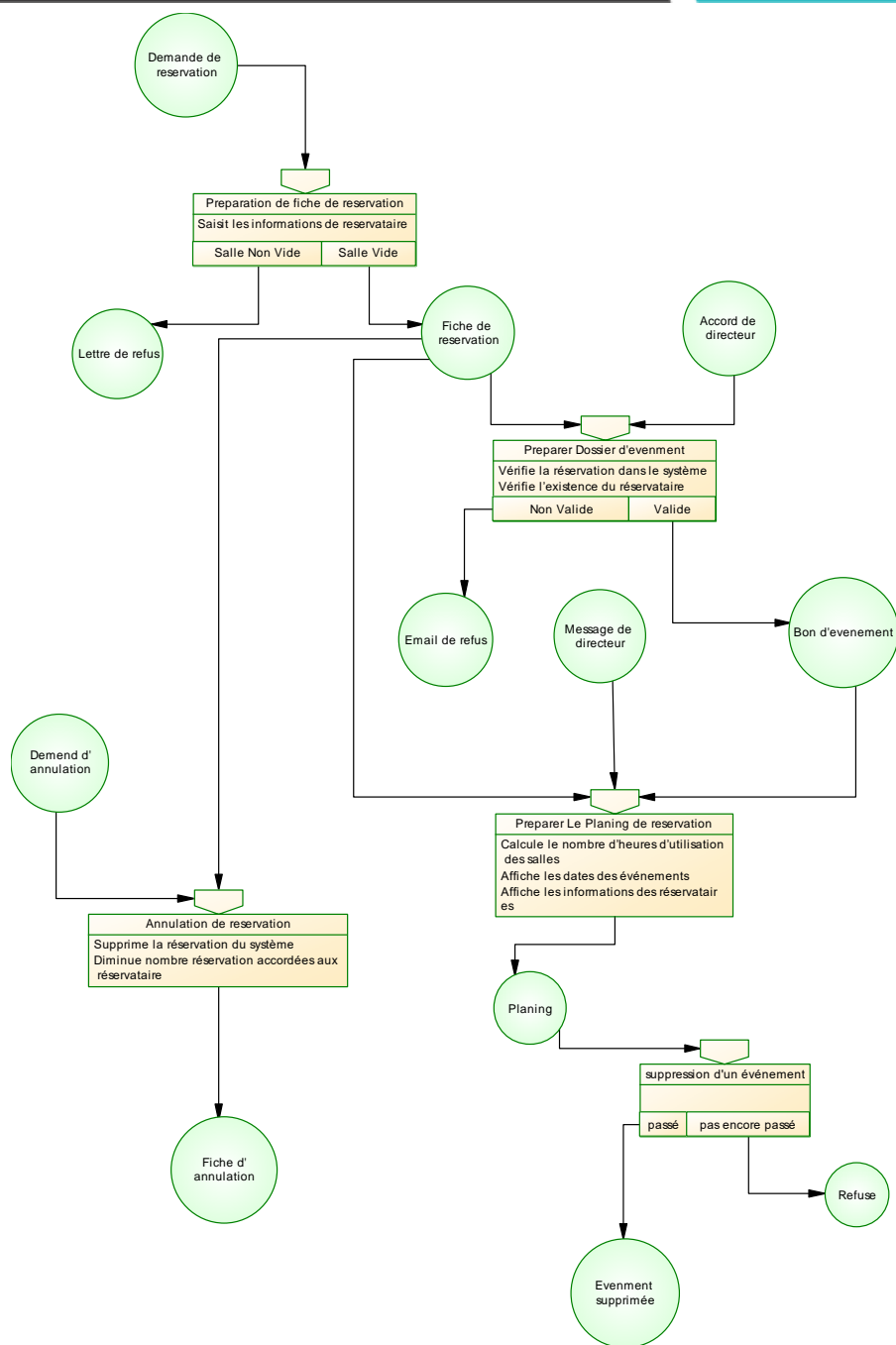
▪ ***Les opérations du système :***

- * Préparation de fiche de réservation
- * Préparation du dossier d'événement
- * Annulation de la réservation
- * Préparation du Planning des réservations
- * Suppression de la réservation

▪ ***Modèle conceptuel de traitement MCT :***

Schéma du module conceptuel de traitement :







➤ Modèle organisationnel de traitement MOT :

Le modèle organisationnel des traitements s'attache à décrire les propriétés des traitements non traitées par le modèle conceptuel des données, c'est-à-dire :

- * Le temps *Quand exécute-t-on l'opération ?*
- * Le lieu *Qui exécute ? Acteurs*
- * La Nature d'opération *Manuelle ou Automatique*

$$\text{MOT} = \text{MCT} + \text{Temps} + \text{Lieu} + \text{Nature}$$

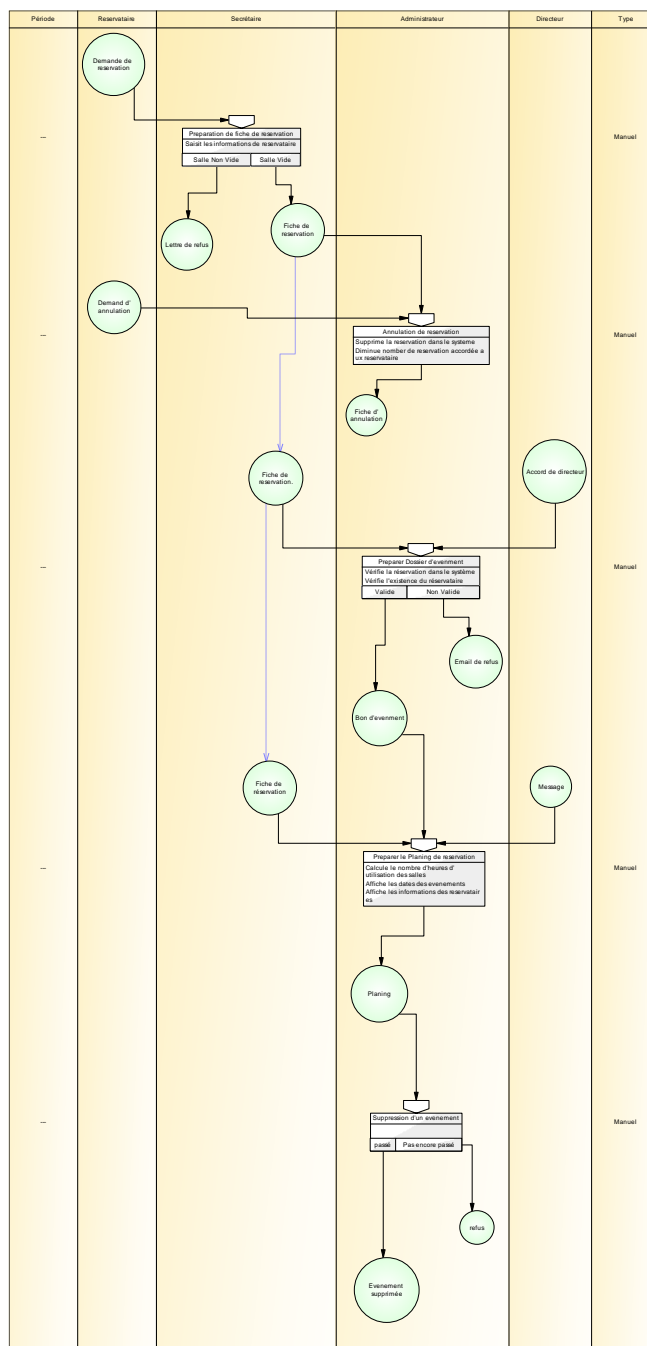
▪ **Les Acteurs :**

- * Réservataires
- * Secrétaire
- * Administrateur
- * Directeur

NB : Tous les opérations sont manuelles

NB : Voir le Modèle dans la page suivante. C'est claire juste fait un zoom.







Base de Données :

Une base de données, **BDD** ou encore **database** en anglais, regroupe un ensemble d'informations qui est organisé pour être accessible, géré et mis à jour facilement par ses utilisateurs. Les données sont organisées en lignes, colonnes et tableaux et indexées pour faciliter les recherches.

Dans Tous Le Projet en se basée sur les données des tableaux suivante :

ETUDIANT						
IdEtudaint	Nom	Prenom	Age	Ville	DateNaissance	Filière
1	Mesrar	Hamza	21	CasaBlanca	21/11/2000	IID1
2	Mbrouk	Rida	21	Kalaat Mgouna	3/10/2000	IRIC1
3	Najid	Mohammed	22	Casablanca	4/6/1999	IID2
4	Merkhas	Hamza	20	Khouribga	2/5/2001	GI1
6	Mabrouk	Kamal	21	Berrechid	5/9/2000	GI2
7	Kerboub	Hamza	19	Berrechid	14/05/2002	API2

PROFESSEUR			
IdProfesseur	Nom	Prenom	Departement
1	Gherabi	Noreddin	Math info
2	Khalfi	Hamza	Math Info
3	Maleh	Yassin	Réseaux et Télécom
4	Abid	Aziza	Génie des procédés
5	Maaidar	Kamal	Electrique
6	Ghezdali	Abdelghani	Math info
7	Mounir	Soufyane	Réseaux et Télécom

ADMINISTRATEUR			
IdAdmin	Nom	Prenom	Fonction
1	admin1	admin1	Fonc1
2	admin2	admin2	Fonc2
3	admin3	admin3	Fonc3



EQUIPEMENT				
NumSerie	#NumSalle	Type	Fabricant	Number
111	11	Data Show	Epson	1
111	12	Data Show	Epson	1
111	21	Data Show	Benq	1
111	22	Data Show	Epson	1
111	31	Data Show	Benq	1
111	32	Data Show	Benq	1
111	33	Data Show	Epson	1
222	11	Table	Kitea	16
222	12	Table	Green chair	16
222	22	Table	Green chair	16
222	23	Table	Kitea	26
333	13	chaise	Kitea	60
333	23	chaise	Kitea	80
333	31	chaise	Green chaire	80
333	33	chaise	Green chair	60
333	42	chaise	Kitea	60
333	51	chaise	Green chair	60

SALLES		
NumSalle	Capacite	#CodeBloc
11	50	A
12	50	A
13	50	A
21	50	B
22	50	B
23	70	B
31	50	C
32	50	C
33	50	C
41	50	D
42	50	D
43	50	D
51	50	E
52	50	E
53	50	E

BLOC	
CodeBloc	Departement
A	Electrique
B	Réseaux et Télécom
C	Math Info
D	Génie des procédés
E	Génie des procédés

EVENEMENTS						
IdEvenement	Sujet	DateRealisation	#NumSalle	#IdEtudiant	#IdProfesseur	IdAdmin
1	IA	9/11/2021	23	1		
2	BigData	16/11/2021	33		1	
3	Integration	20/12/2021	13			2
4	OP	25/12/2021	11		2	
5	CyberSecurity	2/12/2021	32	2		



RESERVATIONS					
NumReservation	DateReservation	#IdProfesseur	#IdEtudiant	#IdAdmin	#NumSalle
1	5/11/2021		1		23
2	9/11/2021	1			33
3	15/11/2021	3			21
4	23/11/2021		2		32
5	2/12/2021		7		12
6	12/12/2021			2	13
7	16/12/2021	2			11
8	24/12/2021	1			32
9	4/1/2022		1		32

➤ Les requêtes SQL en langage LCD :

☑ Création des Espaces :

Dans notre cas on va créer 3 *Espaces*

- * Espace Administrateur
- * Espace Professeur
- * Espace Etudiant

❖ *Espace Administrateur (Direction) :*





N'importe qu'elle utilisateur avec l'accès de cette espace a le droit de Réserver un salle, Afficher le planning des salles, Ajouter un évènement, Afficher tous les évènements, Valider les réservations, Valider les évènements.

→ *Création de l'utilisateur Administrateur :*

On générale en utilise la requête suivante :

```
SQL> CREATE USER USER_NAME IDENTIFIED BY USER_PASS;
```

Dans notre Cas.

```
SQL> CREATE USER admin IDENTIFIED BY admin;  
User created.
```

Le principe est de créer un rôle administrateur pour affecter cette rôle aux utilisateurs de la direction.

Pour crée un rôle en générale on utilise la requête suivante :

```
SQL> CREATE ROLE ROLE_NAME;
```





Dans notre Cas :

```
SQL> CREATE ROLE roleAdmins;
```

Les requêtes utilisées pour affecter les droits d'accès :

On génère pour affecter un droit ou une option à un rôle ou à un utilisateur on utilise

Pour un utilisateur :

```
SQL> GRANT Option_Name ON Table_Name TO USER_Name;
```

Pour un Rôle :

```
SQL> GRANT Option_Name ON Table_Name TO ROLE_NAME;
```

Affecte un Rôle à un utilisateur

```
SQL> GRANT ROLE_NAME TO USER_NAME;
```





Les requêtes utiliser pour affectez le droit nécessaire au Rôle roleAdmins sont comme la suivante :

```
SQL> GRANT create session, create user TO roleAdmins;
SQL> GRANT alter session TO roleAdmins;
SQL> GRANT insert,select, update, delete, ON reservation TO roleAdmins;
SQL> GRANT insert, select, update, delete, ON Evenements TO roleAdmins;
SQL> GRANT insert,select,update,delete, ON Administrateur TO roleAdmins;
SQL> GRANT insert, select, update, delete, ON Etudiant TO roleAdmins;
SQL> GRANT insert, select, update, delete, ON Professeur TO roleAdmins;
SQL> GRANT select ON Salles TO roleAdmins;
SQL> GRANT select ON Bloc TO roleAdmins;
SQL> GRANT select ON Equipement TO roleAdmins;
SQL> GRANT roleAdmins TO admin;
SQL> GRANT unlimited tablespace TO admin;
```





❖ *Espace Professeur :*

→ *Création de l'utilisateur Professeur :*

```
SQL> CREATE USER Profs IDENTIFIED BY Profs;  
User created.
```

→ *Création du rôle roleProf :*

```
SQL> CREATE ROLE roleProf;
```

Les requêtes utiliser pour affectez le droit nécessaire au Rôle roleProf sont comme la suivante :

```
SQL> GRANT create session, create user TO roleProf;
```

```
SQL> GRANT alter session TO roleProf;
```

```
SQL> GRANT select ON reservation TO roleProf;
```

```
SQL> GRANT select ON Evenements TO roleProf;
```





```
SQL> GRANT select ON Salles TO roleProf;  
SQL> GRANT select ON Bloc TO roleProf;  
SQL> GRANT select ON Equipement TO roleProf;  
SQL> GRANT roleProf TO Profs;
```

❖ *Espace Etudiants :*

→ *Création de l'utilisateur Etudiants :*

```
SQL> CREATE USER Etudiants IDENTIFIED BY Etd123;  
  
User created.
```

→ *Création du rôle roleEtd :*

```
SQL> CREATE ROLE roleEtd;
```

Les requêtes utiliser pour affectez le droit nécessaire au Rôle roleEtd sont comme la suivante :





```
SQL> GRANT create session, create user TO roleEtd;  
SQL> GRANT alter session TO roleEtd;  
SQL> GRANT select ON reservation TO roleEtd;  
SQL> GRANT select ON Evenements TO roleEtd;  
SQL> GRANT select ON Salles TO roleEtd;  
SQL> GRANT select ON Bloc TO roleEtd;  
SQL> GRANT select ON Equipement TO roleEtd;  
SQL> GRANT roleEtd TO Etudiants;
```

➤ le schéma relationnel (Langage LDD) :

Langage LLD : Langage de Définition de Données ou en anglais DDL : Data Definition Language a but de créer et supprimer des objets dans la base de données (tables, contraintes d'intégrité, vues, etc...).

→ Création des **tables** nécessaires pour ce projet a l'aide de Langage **LLD** :

NB : j'ai créé les constraints au même temps que je crée un table.





Pour crée un table en generale on utilise :

```
SQL> CREATE TABLE TABLE_NAME (field1_name Type Constrain, . . . . . , fieldn_name Type Constraint);
```



Table Etudiant :

```
SQL> create table Etudiant(IdEtudiant number PRIMARY KEY NOT NULL,
  2  Nom varchar(20) NOT NULL,
  3  Prenom varchar(30) NOT NULL,
  4  Age number check(Age>=18) NOT NULL,
  5  Ville varchar(20),
  6  DateNaissance date,
  7  Filier varchar(20) NOT NULL);
```

Table created.

```
SQL> describe Etudiant
```

Name	Null?	Type
-----	-----	-----
IDETUDIANT	NOT NULL	NUMBER
NOM	NOT NULL	VARCHAR2(20)
PRENOM	NOT NULL	VARCHAR2(30)
AGE	NOT NULL	NUMBER
VILLE		VARCHAR2(20)
DATENAISSANCE		DATE
FILIER	NOT NULL	VARCHAR2(20)



**Table Professeur :**

```
SQL> create table Professeur(IdProf number PRIMARY KEY NOT NULL,
  2  Nom varchar(20) NOT NULL,
  3  Prenom varchar(30) NOT NULL,
  4  Departement varchar(30) NOT NULL);
```

Table created.

```
SQL> describe Professeur
```

Name	Null?	Type
-----	-----	-----
IDPROF	NOT NULL	NUMBER
NOM	NOT NULL	VARCHAR2(20)
PRENOM	NOT NULL	VARCHAR2(30)
DEPARTEMENT	NOT NULL	VARCHAR2(30)

**Table Administrateur :**

```
SQL> create table Administrateur(IdAdmin number PRIMARY KEY NOT NULL,
  2  Nom varchar(20) NOT NULL,
  3  Prenom varchar(30) NOT NULL,
  4  Fonction varchar(20) NOT NULL);
```

Table created.





```
SQL> describe Administrateur
```

Name	Null?	Type

IDADMIN	NOT NULL	NUMBER
NOM	NOT NULL	VARCHAR2(20)
PRENOM	NOT NULL	VARCHAR2(30)
FONCTION	NOT NULL	VARCHAR2(20)



Table Salle :

```
SQL> create table Salles(NumSalle number PRIMARY KEY NOT NULL,
  2  Capacite number NOT NULL,
  3  CodeBloc varchar(3),
  4  CONSTRAINT FK_Bloc Foreign Key(CodeBloc)
  5  References Bloc(CodeBloc)
  6  ON DELETE CASCADE);
```

Table created.

```
SQL> describe Salles
```

Name	Null?	Type

NUMSALLE	NOT NULL	NUMBER
CAPACITE	NOT NULL	NUMBER
CODEBLOC		VARCHAR2(3)





Table reservation :

```
SQL> create table reservation(NumReservation number PRIMARY KEY NOT NULL,  
2 DateReservation date NOT NULL,  
3 NumSalle number,  
4 IdEtudiant number,  
5 IdProf number,  
6 IdAdmin number,  
7 CONSTRAINT FK_Salles FOREIGN KEY(NumSalle) REFERENCES Salles(NumSalle)  
8 ON DELETE CASCADE,  
9 CONSTRAINT FK_Admin FOREIGN KEY(IdAdmin) REFERENCES Administrateur(IdAdmin)  
10 ON DELETE CASCADE,  
11 CONSTRAINT FK_Prof FOREIGN KEY(IdProf) REFERENCES Professeur(IdProf)  
12 ON DELETE CASCADE,  
13 CONSTRAINT FK_Etudiant FOREIGN KEY(IdEtudiant) REFERENCES  
Etudiant(IdEtudiant)  
14 ON DELETE CASCADE);
```

Table created.





QL> describe reservation

Name	Null?	Type
-----	-----	-----
NUMRESERVATION	NOT NULL	NUMBER
DATERESERVATION	NOT NULL	DATE
NUMSALLE		NUMBER
IDETUDIANT		NUMBER
IDPROF		NUMBER
IDADMIN		NUMBER



Table Bloc :

SQL> create table Bloc(CodeBloc varchar(3) PRIMARY KEY NOT NULL,
2 Departement varchar(20) NOT NULL);

Table created.

SQL> describe Bloc

Name	Null?	Type
-----	-----	-----
CODEBLOC	NOT NULL	VARCHAR2(3)
DEPARTEMENT	NOT NULL	VARCHAR2(20)





Table Equipement :

```
SQL> create table Equipement(NumSerie number NOT NULL,  
2  NumSalle number NOT NULL,  
3  Type varchar(20) NOT NULL,  
4  Fabricant varchar(20),  
5  Nomber number NOT NULL,  
6  CONSTRAINT FK3_Salles FOREIGN KEY(NumSalle)  
7  REFERENCES Salles(NumSalle)  
8  ON DELETE CASCADE);
```

Table created.

```
SQL> alter table Equipement  
2  ADD CONSTRAINT PK_Equip PRIMARY KEY (NumSerie, NumSalle);
```

Table altered.

```
SQL> describe Equipement
```

Name	Null?	Type

NUMSERIE	NOT NULL	NUMBER
NUMSALLE	NOT NULL	NUMBER
TYPE	NOT NULL	VARCHAR2(20)
FABRICANT		VARCHAR2(20)
NOMBER	NOT NULL	NUMBER





Table Evenements :

```
SQL> create table Evenements(IdEvent number PRIMARY KEY NOT NULL,  
2  Sujet varchar(50) NOT NULL,  
3  DateRealisation date NOT NULL,  
4  NumSalle number,  
5  IdEtudiant number,  
6  IdProf number,  
7  IdAdmin number,  
8  CONSTRAINT FK2_Salles FOREIGN KEY(NumSalle) REFERENCES Salles(NumSalle)  
9  ON DELETE CASCADE,  
10 CONSTRAINT FK2_Etudiant FOREIGN KEY(IdEtudiant) REFERENCES Etudiant(IdEtudiant)  
11 ON DELETE CASCADE,  
12 CONSTRAINT FK2_Prof FOREIGN KEY(IdProf) REFERENCES Professeur(IdProf)  
13 ON DELETE CASCADE,  
14 CONSTRAINT FK2_Admin FOREIGN KEY(IdAdmin) REFERENCES Administrateur(IdAdmin)  
15 ON DELETE CASCADE);
```

Table created.





```
SQL> describe Evenements
```

<i>Name</i>	<i>Null?</i>	<i>Type</i>
-----	-----	-----
<i>IDEVENT</i>	<i>NOT NULL</i>	<i>NUMBER</i>
<i>SUJET</i>	<i>NOT NULL</i>	<i>VARCHAR2(50)</i>
<i>DATEREALISATION</i>	<i>NOT NULL</i>	<i>DATE</i>
<i>NUMSALLE</i>		<i>NUMBER</i>
<i>IDETUDIANT</i>		<i>NUMBER</i>
<i>IDPROF</i>		<i>NUMBER</i>
<i>IDADMIN</i>		<i>NUMBER</i>

➤ Langage de Manipulation de donne LMD:

Ce langage permet de manipuler les données de la base et de les mettre à jour. Comme l'insertion Des données modifier le données et la suppression du données, et affichage (Sélection) du donnée.

→ *L'insertion des données Remplissage des tables :*

Entrer les donnée dans chaque table de système :

▶ **Table Etudiant :**





```
SQL> insert into Etudiant(IdEtudiant,Nom,Prenom,Age,Ville,DateNaissance,Filier)
  2  VALUES(1, 'Mesrar', 'Hamza', 21, 'Casa Blanca', TO_DATE('21/11/2000', 'DD/MM/YYYY'), 'IID1');
1 row created.
```

```
SQL> select * from Etudiant;
```

IDETUDIANT	NOM	PRENOM	AGE	VILLE	DATENAISS	FILIER
1	Mesrar	Hamza	21	Casa Blanca	21-NOV-00	IID1

► Table Professeur :

```
SQL> insert into Professeur(IdProf,Nom,Prenom,Departement)
  2  VALUES(1, 'Gherabi', 'Noreddin', 'Math Info');

1 row created.
```

```
SQL> select * from Professeur;
```

IDPROF	NOM	PRENOM	DEPARTEMENT
1	Gherabi	Noreddin	Math Info

► Table Administrateur :





```
SQL> insert into Administrateur(IdAdmin,Nom,Prenom,Fonction)
```

```
2 VALUES(1,'admin1','admin2','Fonc1');
```

```
1 row created.
```

```
SQL> select * from Administrateur;
```

IDADMIN	NOM	PRENOM	FONCTION
---------	-----	--------	----------

1	admin1	admin2	Fonc1
---	--------	--------	-------

► Table reservation :

```
SQL> insert into reservation(NumReservation,DateReservation,NumSalle,IdEtudiant,IdProf,IdAdmin)
```

```
2 VALUES(1,TO_DATE('05/11/2021','DD/MM/YYYY'),23,1,NULL,NULL);
```

```
1 row created.
```

```
SQL> select * from reservation;
```

NUMRESERVATION	DATERESER	NUMSALLE	IDETUDIANT	IDPROF	IDADMIN
----------------	-----------	----------	------------	--------	---------

1	05-NOV-21	23	1		
---	-----------	----	---	--	--

► Table Salles :





```
SQL> insert into Salles(NumSalle,Capacite,CodeBloc)
```

```
2 VALUES(11,50,'A');
```

```
1 row created.
```

```
SQL> select * from Salles;
```

```
NUMSALLE  CAPACITE  COD
```

```
-----
```

```
11          50  A
```



Table Equipement :

```
SQL> insert into Equipement(NumSerie, NumSalle, Type, Fabricant, Number)
```

```
2 VALUES (111, 11, 'Video Projector', 'Epson',1);
```

```
1 row created.
```

```
SQL> select * from Equipement;
```

```
NUMSERIE  NUMSALLE  TYPE                FABRICANT                NUMBER
```

```
-----
```

```
111        11 Video Projector  Epson                    1
```



Table Bloc :



```
SQL> insert into Bloc(CodeBloc, Departement)
  2  VALUES ('A','Genie Electrique');
1 row created.
```

```
SQL> select * from Bloc;
```

```
COD DEPARTEMENT
```

```
-----
```

```
A   Genie Electrique
```

▶ Table Evenements :

```
SQL> insert into Evenements(IdEvent, Sujet, DateRealisation, NumSalle, IdEtudiant, IdProf, IdAdmin)
  2  VALUES (1,'IA', TO_DATE('09/11/2021','DD/MM/YYYY'), 23, 1, NULL, NULL);
1 row created.
```

```
SQL> select * from Evenements;
```

```
  IDEVENT  SUJET    DATEREALI  NUMSALLE  IDETUDIANT    IDPROF    IDADMIN
-----
```

1	IA	09-NOV-21	23	1		
---	----	-----------	----	---	--	--

NB : On remplirai les tables jusqu'à obtienne la résultat dans les tableaux mentionné dans la premier fois.





→Affichage Ou La Selection des données : (Exemple de Selection)

▣ affichage de la liste des salles réservées par un professeur

```
SQL> select * from Salles
      2 where NumSalle in(select NumSalle from reservation
      3 where IdProf in(select IdProf from Professeur where Nom='Gherabi'));
```

NUMSALLE	CAPACITE	COD
33	50	C
32	50	C

▣ Affichage de la liste des salles réservées par un Etudiant

```
SQL> select * from Salles
      2 where NumSalle in(select NumSalle from reservation
      3 where IdEtudiant in(select IdEtudiant from Etudiant where Nom='Mesrar'));
```

NUMSALLE	CAPACITE	COD
23	70	B
32	50	C





► Affichage tous les reservation du Profs de l'année 2021

```
SQL> select * from reservation
2 where EXTRACT(YEAR FROM TO_DATE(DateReservation, 'DD/MM/YYYY'))=21
3 AND IdProf IS NOT NULL;
```

NUMRESERVATION	DATERESER	NUMSALLE	IDETUDIANT	IDPROF	IDADMIN
2	03-NOV-21	33		1	
3	15-NOV-21	21		3	
7	16-DEC-21	11		2	
8	24-DEC-21	32		1	

► Affichage Tous les evenements realisé par les etudiants dans le moins Novembre de 2021

```
SQL> select * from Evenements
2 where EXTRACT(YEAR FROM TO_DATE(DateRealisation, 'DD/MM/YYYY'))=21
3 AND EXTRACT(MONTH FROM TO_DATE(DateRealisation, 'DD/MM/YYYY'))=11
4 AND IdEtudiant IS NOT NULL;
```

IDEVENT	SUJET	DATEREALI	NUMSALLE	IDETUDIANT	IDPROF	IDADMIN
1	IA	09-NOV-21	23	1		





→ Afficher la list des salle ainsi que votre date de reservation et Nome complet de Prof reserver just de la moins Decembre 2021

```
SQL> select numSalle AS SalleNumero, (Nom || ' ' || Prenom) as ReserverPar, DateReservation As En from reservation, Professeur
2 where EXTRACT(YEAR FROM TO_DATE(DateReservation, 'DD/MM/YYYY'))=21 AND EXTRACT(MONTH FROM
TO_DATE(DateReservation, 'DD/MM/YYYY'))=12
3 AND reservation.IdProf IS NOT NULL AND reservation.IdProf=Professeur.IdProf;
```

SALLENUMERO	RESERVERPAR	EN
11	Khalfi Hamza	16-DEC-21
32	Gherabi Noredin	24-DEC-21



Afficher La list des salle qui sont jamais reserver

```
SQL> select NumSalle, Salles.CodeBloc, Departement from Salles, Bloc
2 where NumSalle NOT IN(select NumSalle from reservation)
3 AND Salles.CodeBloc=Bloc.CodeBloc;
```

NUMSALLE	COD	DEPARTEMENT	NUMSALLE	COD	DEPARTEMENT
22	B	Resaux et Telecom	41	D	Genie procodes
31	C	Math Info	51	E	Genie procodes
42	D	Genie procodes	52	E	Genie procodes
43	D	Genie procodes	53	E	Genie procodes

8 rows selected.





- Afficher la list des salles qui contien un video projecteur de marque Epson et plus de 15 Table de marque Kitea

```
SQL> select NumSalle,Salles.CodeBloc,Departement from Salles,Bloc
2  where NumSalle in(select NumSalle from Equipement where Fabricant='Epson'
3  INTERSECT select NumSalle from Equipement where Type='Table'
4  AND Fabricant='Kitea' AND Number>15)
5  AND Salles.CodeBloc=Bloc.CodeBloc;
```

NUMSALLE COD DEPARTEMENT

```
-----
11 A    Genie Electrique
```

- Afficher le plannig du évènements :

```
SQL> select (Salles.NumSalle || ' Bloc ' || CodeBloc) as Place,Sujet AS Sujet_De_Evenement,DateRealisation
2  from Evenements,Salles where Salles.NumSalle=Evenements.NumSalle
3  ORDER BY DateRealisation ASC;
```

Le résultat est comme la suivante :





PLACE	SUJET_DE_EVENEMENT	DATEREALI

23 Bloc B	IA	09-NOV-21
33 Bloc C	Big Data	16-NOV-21
32 Bloc C	Cyber Security	02-DEC-21
13 Bloc A	Integration	20-DEC-21
11 Bloc A	OP	25-DEC-21

→ *Suppression des données*

En generale On utilise la requete suivante :

SQL> delete from *TABLE_NAME* where *CONDITION*;

▢ Supprimer un reservation

```
SQL> delete from reservation
      2 where NumReservation=9;
1 row deleted.
```

▢ Supprimer un Evénement

```
SQL> delete from Evenements where IdEvent=3;
1 row deleted.
```





Le Même principe répète pour supprimer n'importe qu'elle données.

→ *Modification des données*

▶ En generale On utilise la requete suivante :

```
SQL> UPDATE TABLE_NAME SET COLUMN_NAME=VALUES WHERE CONDITION;
```

Conclusion :

Grâce à ce projet j'ai acquis plusieurs méthodes et techniques pour la gestion de base données et de concevoir un système informatique comme un système de gestion de la salle et l'organisation des événements de notre école qui je souhaite d'être appliquée aux futures, et finalement j'ai remercié, que vous nous donner la chance d'appliquer tous qu'on a étudiés dans ce module, et d'améliorer nos compétences.

FIN !

