

Quelques requetes SQL :

1/Requête pour vérifier si un bureau est réservé ou non :

1ere version :

```
SELECT CASE
  WHEN EXISTS (
    SELECT *
    FROM reservations r ,bureau b
    WHERE b.id_bureau = r.id_bureau and b.nom='bx23'
    AND '2023-05-29 14:00:00 ' BETWEEN r.date_debut and r.date_fin
  ) THEN 'Réservé'
  ELSE 'Non réservé'
END AS état
FROM Bureau
WHERE nom='bx23';
```

2eme version : en disant juste que la date fin de réservation est supérieur à la date actuelle

```
SELECT CASE
  WHEN EXISTS (
    SELECT *
    FROM reservations r ,bureau b
    WHERE b.id_bureau = r.id_bureau and b.nom='bx24'
    AND Date_fin > NOW()
  ) THEN 'Réservé'
  ELSE 'Non réservé'
END AS état
FROM Bureau
WHERE nom='bx24';
```

2_1/bureau est t-il il reservable

```
SELECT id_bureau
from bureau
where nom='bx24' and reservable = 1;
```

2_2/si oui est t-il dans une réservation

```
SELECT r.id_bureau,r.id_reservation
from reservations r
join bureau b on b.id_bureau=r.id_bureau
where b.nom='bx23' and b.reservable=1;
```

2_3/savoir si la réservation est confirmé

```
select r.id_bureau ,r.id_reservation
from reservations r
join bureau b on b.id_bureau= r.id_bureau
where confirmation =1 and nom='bx23';
```

2/l'état d'un ensemble de bureaux à un instant donné :

```
SELECT b.nom,
CASE
    WHEN r.id_bureau IS NOT NULL THEN 'Réservé'
    ELSE 'Non réservé'
END AS état
FROM bureau b
LEFT JOIN reservations r ON b.id_bureau = r.id_bureau
AND '2023-05-29 14:00:00 ' BETWEEN r.date_debut AND r.date_fin; //ou on change par
Date_fin > NOW()
```

3/Créer une réservation d'un bureau

```
INSERT INTO reservations ( id_bureau, date_reservation,id_pers_create,id_beneficiaire ,id_salle)
VALUES ( 12, '2023-06-01' ,1,1,5) ;
SELECT LAST_INSERT_ID() AS id_reservation;
```

4/modifier une réservation d'un bureau

```
UPDATE reservations
```

```
SET date_debut = '2023-06-02 10:00:00', -- Nouvelle date de début
```

```
    date_fin = '2023-06-02 18:00:00' -- Nouvelle date de fin
```

```
WHERE id_bureau = 12 -- Identifiant du bureau à modifier
```

```
    AND id_reservation = 10; -- Identifiant de la réservation à modifier
```

5/supprimer une réservation :

```
DELETE FROM reservations
```

```
where id_reservation = 11; -- Identifiant de la réservation à supprimer
```

6/ Afficher les détails associés à une réservation donnée, formulée par le demandeur :

```
SELECT r.id_reservation, r.date_reservation, r.date_debut, r.date_fin, r.type, r.description,
```

```
    b.nom AS nom_bureau, u.nom AS nom_demandeur, u.email AS email_demandeur ,p.nom as
nom_personne ,p.prenom as prenom_personne ,p.statut as statut_personne,s.nom as nom_salle
```

```
FROM reservations r
```

```
JOIN bureau b ON r.id_bureau = b.id_bureau
```

```
JOIN utilisateur u ON r.id_pers_create = u.id_utilisateur
```

```
JOIN personne p ON r.id_pers_create = p.encadrant
```

```
JOIN salles s ON r.id_salle = s.id_salle
```

```
WHERE r.id_reservation = 2; -- Identifiant de la réservation spécifique
```

7/affecter plusieurs personne a un bureau à la main

```

INSERT INTO affectations (id_bureau)
VALUES (12);

UPDATE affectations SET id_affectation = @id_affectation WHERE id_bureau = 12;

INSERT INTO affectation_personne (id_affectation, id_personne)
VALUES
    (@id_affectation, 1),
    (@id_affectation, 2);

```

8/afficher l'ensemble des personnes affectées à un bureau (si bureau partagé) à une date donnée : ceci marche que si il y a une affectation de plusieurs personnes à un bureau à la main (avec insert to)

```

SELECT DISTINCT p.* FROM personne p
JOIN affectation_personne a ON p.id_personne = a.id_personne
left JOIN affectations f ON a.id_affectation = f.id_affectation
left join reservations r on r.id_bureau = f.id_bureau

WHERE r.date_debut <= '2023-06-01'
AND r.date_fin >= '2023-06-01'
AND f.id_bureau = 12;

```

9/L'ensemble des personnes affectées à un bureau à un instant donné

```

select p.*
from personne p
join reservations r on r.id_beneficiaire = p.id_personne
WHERE r.date_debut <= '2023-06-01'
AND r.date_fin >= '2023-06-01'
AND id_bureau = 12;

```

10/Afficher l'historique des réservations pour un bureau donné

```
SELECT DISTINCT r.* ,p.nom as nom_demandeur,p.prenom as prenom_demandeur ,u.nom as  
nom_personne , u.prénom as prenom_personne  
  
FROM reservations r  
  
JOIN personne P ON P.id_personne =r.id_beneficiaire  
  
join utilisateur u on r.id_pers_create =u.id_utilisateur  
  
WHERE id_bureau = 12  
  
ORDER BY date_reservation DESC;
```

11/Afficher l'historique des réservations pour un donné

```
SELECT DISTINCT r.* ,p.nom as nom_demandeur,p.prenom as prenom_demandeur ,u.nom as  
nom_personne , u.prénom as prenom_personne  
  
FROM reservations r  
  
JOIN personne P ON P.id_personne =r.id_beneficiaire  
  
join utilisateur u on r.id_pers_create =u.id_utilisateur  
  
WHERE id_salle = 6  
  
ORDER BY date_reservation DESC;
```

12/Afficher l'historique des réservations pour un bureau donné et une salle donnée

```
SELECT DISTINCT r.* ,p.nom as nom_demandeur,p.prenom as prenom_demandeur ,u.nom as  
nom_personne , u.prénom as prenom_personne  
  
FROM reservations r  
  
JOIN personne P ON P.id_personne =r.id_beneficiaire  
  
join utilisateur u on r.id_pers_create =u.id_utilisateur  
  
WHERE id_bureau = 14  
  
and id_salle= 5  
  
ORDER BY date_reservation DESC;
```

13/Créer un bureau

```
INSERT INTO bureau (nom, capacité,reservable,etat,partageable, id_salle)
```

```
SELECT 'Bureau b', 3,1,'libre',1, id_salle
```

```
FROM salles
```

```
WHERE nom = 's1';
```

14/Modifier un bureau

```
UPDATE bureau AS b
```

```
SET b.nom ='bureau c',b.capacité = 8 ,b.reservable =0, b.etat ='occupé',b.partageable =0 , b.id_salle =  
(
```

```
SELECT id_salle
```

```
FROM salles
```

```
WHERE nom = 's1'
```

```
)
```

```
WHERE b.id_bureau = 41;
```

15/Supprimer un bureau

```
DELETE FROM bureau
```

```
WHERE nom = 'bureau c';
```

15_1/Créer un utilisateur cad le demandeur

```
INSERT INTO utilisateur (nom, prénom, email,mot_passe)
```

```
VALUES ('Doe', 'John', 'shdhed@','dhjjeid');
```

15_2/Créer une personne cad la personne qui sera intégré

```
INSERT INTO personne (nom, prenom, titre, sujet, financement, date_debut, date_fin, encadrant,  
statut)
```

```
SELECT 'Marie', 'Jo', 'projet', 'web', 0, '2023-06-01', '2023-06-03', id_utilisateur, 'stage'
```

```
FROM utilisateur
```

```
WHERE nom = 'Fred' AND prénom = 'daven';
```

16_1/Modifier une personne (la personne qui sera intégré)

UPDATE personne p

JOIN utilisateur u ON p.encadrant = u.id_utilisateur

SET p.nom = 'Nouveau nom', p.prenom = 'Nouveau prénom', p.titre = 'Nouveau titre', p.sujet = 'Nouveau sujet', p.financement = 1, p.date_debut = '2023-06-01', p.date_fin = '2023-06-03', p.statut = 'stage',p.encadrant='1'

WHERE p.id_personne = 5;

16_2/Modifier un utilisateur cad le demandeur

UPDATE utilisateur

SET nom = 'Nouveau nom', prénom = 'Nouveau prénom', email='hdhjzdh@',mot_passe='jhsdh'

WHERE id_utilisateur = 5;

17_1/Supprimer une personne (la personne qui est intégré)

DELETE FROM personne

WHERE id_personne = '4';

17_2/Supprimer un utilisateur cad un demandeur

DELETE FROM utilisateur

WHERE id_utilisateur = '5';

18/Valider une réservation

UPDATE reservations

SET confirmation = 1, date_confirmation = CURDATE()

WHERE id_reservation = '1';