

# Concurrence d'accès.

**Vous copiez les tables t\_employe, t\_service, t\_projet, t\_travail du schéma licpro.**

Donc :

```
create table employe as select * from licpro.t_employe;
create table service as select * from licpro.t_service;
create table projet as select * from licpro.t_projet;
create table travail as select * from licpro.t_travail;
```

Vous ouvrez **deux fenêtres** pour avoir deux transactions en concurrence.

Vous faites ce qui est indiqué ci dessous. Vous faites un compte-rendu où vous dites ce que vous voyez, ce qui se passe, etc. et où vous expliquez pourquoi.

## **1 – LECTURE/ECRITURE.**

### **1.1 – Fonctionnement par défaut.**

#### **1.1.1 – Que voit-on?**

##### **fenêtre1**

```
select * from employe;
```

```
select * from employe;
```

```
select * from employe;
```

##### **fenêtre2**

```
insert into employe values(99,...);
```

```
select * from employe;
commit;
```

#### **1.1.2 – On ne voit pas toujours la même chose dans une transaction.**

##### **fenêtre1**

```
select * from employe;
```

```
select * from employe;
```

```
select * from employe;
```

##### **fenêtre2**

```
delete from employe where nuempl=99;
```

```
select * from employe;
commit;
```

## **1.2 - Transaction read only.**

### **1.2.1 – On voit toujours la même chose .**

#### **fenêtre1**

```
set transaction read only;  
select * from employe;  
  
select * from employe;  
  
select * from employe;  
  
select * from employe;  
  
select * from employe;  
commit;  
select * from employe;
```

#### **fenêtre2**

```
insert into employe values(99,...);  
  
commit;  
  
delete from employe where nuempl=99;  
  
commit;
```

## **2 – ECRITURE/ECRITURE.**

### **2.1 – Fonctionnement par défaut**

#### **2.1.1 – Parfois on croit faire quelque chose et on ne fait rien.**

#### **Fenêtre1**

```
insert into employe values (99,...);  
  
commit;
```

#### **fenêtre2**

```
update employe set ... where nuempl=99;  
  
commit;  
select * from employe;
```

#### **2.1.2 – Ca verrouille.**

#### **fenêtre1**

```
delete from employe where nuempl=99;  
  
commit;
```

#### **fenêtre2**

```
update employe set ... where nuempl=99;  
  
commit;
```

```
select * from employe where nuempl=99;
```

#### **2.1.3 – parfois ça coince.**

#### **fenêtre1**

```
update travail set duree=5 where nuempl=23 and nuproj=135;  
update travail set duree=7 where nuempl=39 and nuproj=135;  
update travail set duree=7 where nuempl=23 and nuproj=135;  
update travail set duree= 5 where nuempl=39 and nuproj=135;  
rollback;
```

#### **fenêtre2**

```
select * from travail where nuproj=135;  
rollback;
```

### 2.1.4 – avec les contraintes de type références il y a un problème.

On s'intéresse à la contrainte travail.nuempl references employe, qui pour le moment n'est pas décrite.

Les select... sont là pour "voir" si la transaction "voit" la contrainte vérifiée ou non.

#### fenêtre1

```
delete from employe where nuempl=17;

select * from travail where nuempl=17;

commit;
```

#### fenêtre2

```
insert into travail values (17,135,5);

select * from employe where nuempl=17;
commit;
```

```
select * from employe where nuempl=17;
select * from travail where nuempl=17;
```

### 2.1.5 – si la contrainte est décrite Oracle pose des verrous et ça marche.

```
alter table travail add constraint trav_empl foreign key (nuempl) references employe;
delete from travail where nuempl=17;
alter table travail add constraint trav_empl foreign key (nuempl) references employe;
insert into employe values (17,...);
commit;
```

#### fenêtre1

```
delete from employe where nuempl=17;

commit;
```

#### fenêtre2

```
insert into travail values (17,135,5);

rollback;
```

## 2.2– Verrouillage explicite.

### 2.2.1 – on fait comme Oracle, pour une table..

```
insert into projet values (222,...);
commit;
```

On s'intéresse à la contrainte, non décrite, travail.nuproj references projet.

#### Fenêtre1

```
lock table travail in share mode;

delete from projet where nuproj=222;
commit;
```

#### fenêtre2

```
insert into travail values (23,222,5);

select * from projet where nuproj=222;
rollback;
```

### 2.2.2 – ... et pour l'autre.

```
insert into projet values (222,...);  
commit;
```

#### fenêtre1

```
delete from projet where nuproj=222;  
  
select * from travail where nuproj=222;  
rollback;
```

#### fenêtre2

```
lock table projet in share mode;  
  
insert into travail values (23,222,5);  
commit;
```

### 2.2.3 – Si la contrainte est vérifiée par trigger...

```
select hebdo, (select sum(duree) from travail  
               where nuempl=employe.nuempl) duree  
from employe where nuempl=19
```

On s'intéresse à la contrainte  $\text{sum(duree)} < \text{hebdo}$ . Les select ... sont là pour "voir" comme le trigger, et donc savoir s'il lève une exception ou non.

#### fenêtre1

```
insert into travail values (19,370,10);  
  
select hebdo from employe where nuempl=19;  
  
commit;
```

#### fenêtre2

```
update employe set hebdo=20 where nuempl=19;  
select sum(duree) from travail where nuempl=19;  
commit;
```

```
select hebdo, (select sum(duree) from travail  
               where nuempl=employe.nuempl) duree  
from employe where nuempl=19;  
delete from travail where nuempl=19 and nuproj=370;  
update employe set hebdo=30 where nuempl=19;  
commit;
```

### 2.2.4 – on doit verrouiller.

```
select hebdo, (select sum(duree) from travail  
               where nuempl=employe.nuempl) duree  
from employe where nuempl=19;
```

#### fenêtre1

```
lock table employe in share mode;  
insert into travail values(19,370,10);  
  
select sum(duree) from travail where nuempl=19;  
commit;
```

#### fenêtre2

```
lock table travail in share mode;  
  
update employe set hebdo=20 where nuempl=19;
```

```
select sum(duree) from travail where nuempl=19;
rollback;
```

```
delete from travail where nuempl=19 and nuproj=370;
commit;
```

### **2.2.5 – Attention, il faut aussi protéger les calculs.**

#### **fenêtre1**

```
insert into travail values(19,370,10);
```

```
select sum(duree) from travail where nuempl=19;
```

```
commit;
```

#### **fenêtre2**

```
insert into travail values(19,492,10);
```

```
select sum(duree) from travail where nuempl=19;
commit;
```

```
select hebdo, (select sum(duree) from travail
               where nuempl=employe.nuempl) duree
from employe where nuempl=19;
```

```
delete from travail where nuempl=19 and nuproj=370;
```

```
delete from travail where nuempl=19 and nuproj=492;
commit;
```

### **2.2.6 – On le fait.**

#### **Fenêtre1**

```
lock table travail in share row exclusive mode;
```

```
insert into travail values(19,370,10);
```

```
select sum(duree) from travail where nuempl=19;
```

```
commit;
```

#### **fenêtre2**

```
lock table travail in share row exclusive mode;
```

```
insert into travail values(19,492,10);
```

```
select sum(duree) from travail where nuempl=19;
rollback;
```