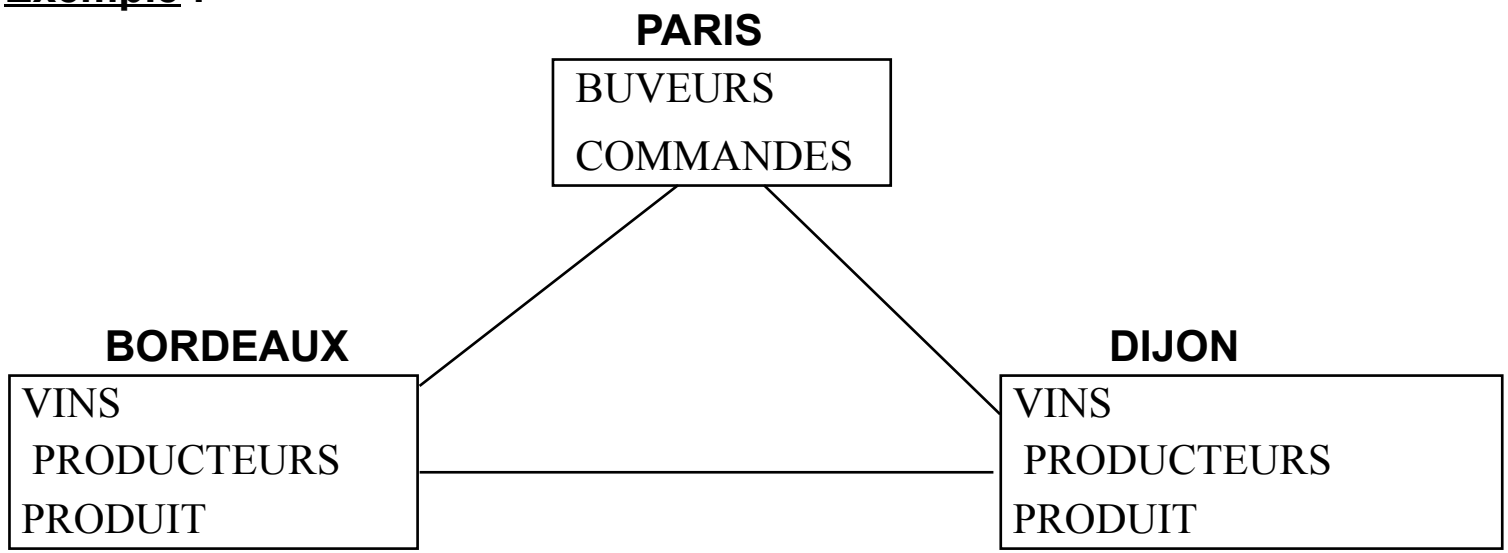
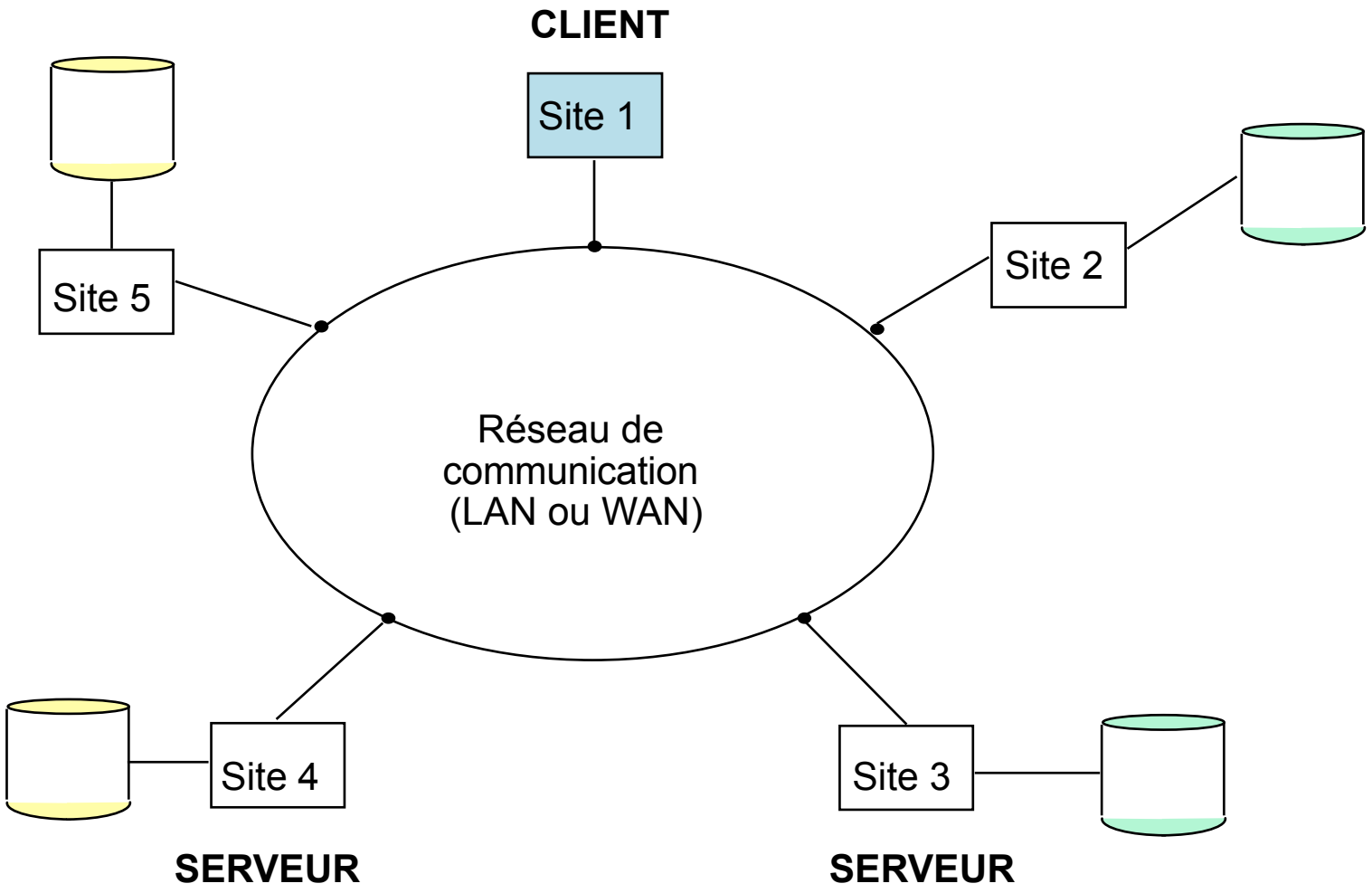
Bases de Données Réparties

- **Définition :**
  - Ensemble de bases de données gérées par des sites différents et apparaissant à l'utilisateur comme une base unique.

Exemple :



La répartition géographique



Objectifs

- **Transparence à la localisation**
  - évaluation et optimisation de requêtes réparties
- **Gestion de transactions réparties**
  - validation atomique répartie
  - contrôle de concurrence réparti
- **Disponibilité**
  - gestion de copies multiples
- **Intégration de bases de données hétérogènes**
  - intégration de bases locales existantes et autonomes

Transparence à la localisation

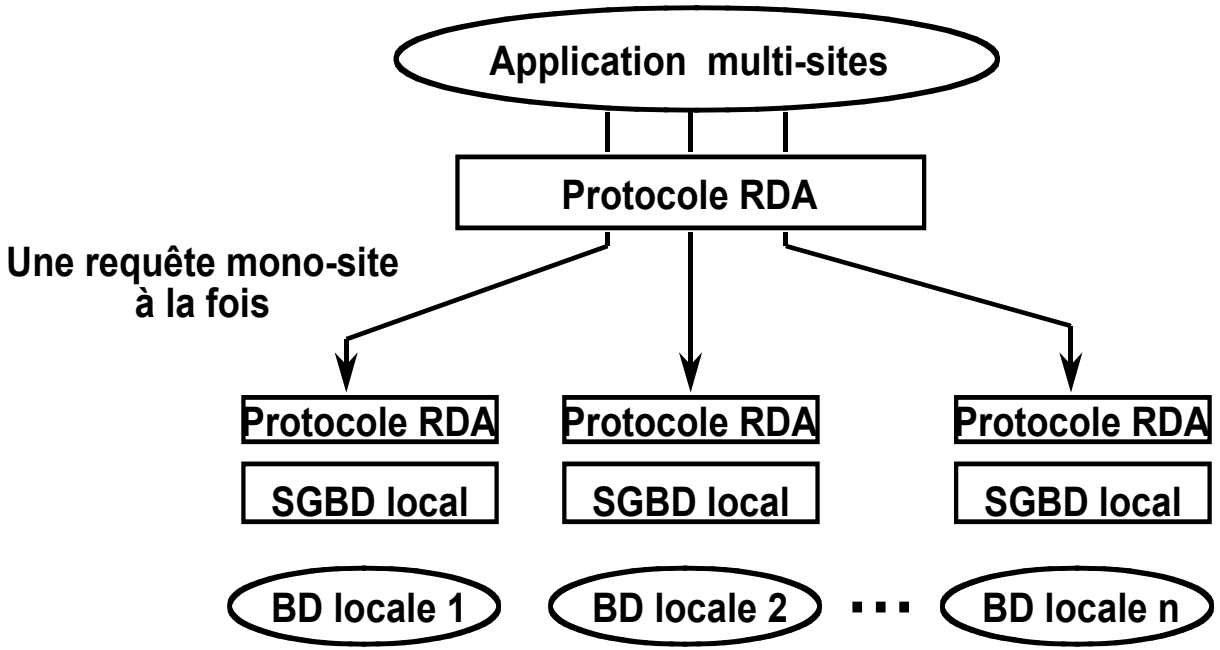
- **Niveau 1 : Client / Multi-Serveur**
- **Niveau 2 : Vues réparties**
- **Niveau 3 : SGBD réparti**
- **Niveau 4 : SGBD Fédéré**

Niveau 1 : Client/Multi-Serveurs

- **Les standards:**
  - RDA (Remote Data Access) de l'ISO
  - SQL-CLI (Client Level Interface) de X/Open

- **Les produits:**
  - ODBC (Open Database Connectivity) de Microsoft
  - IDAPI de Borland, DRDA d'IBM, ...

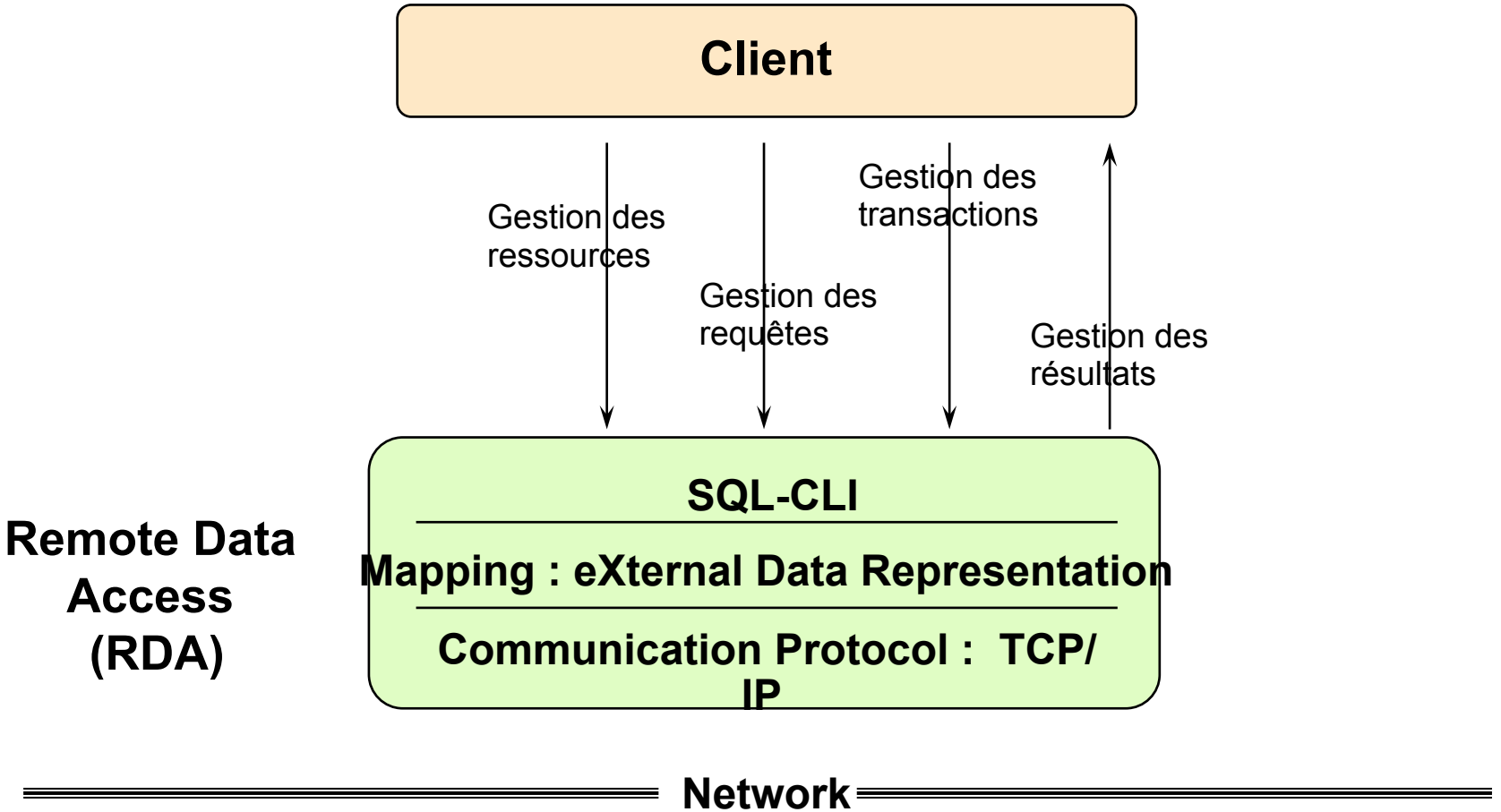
Solution RDA : Caractéristiques



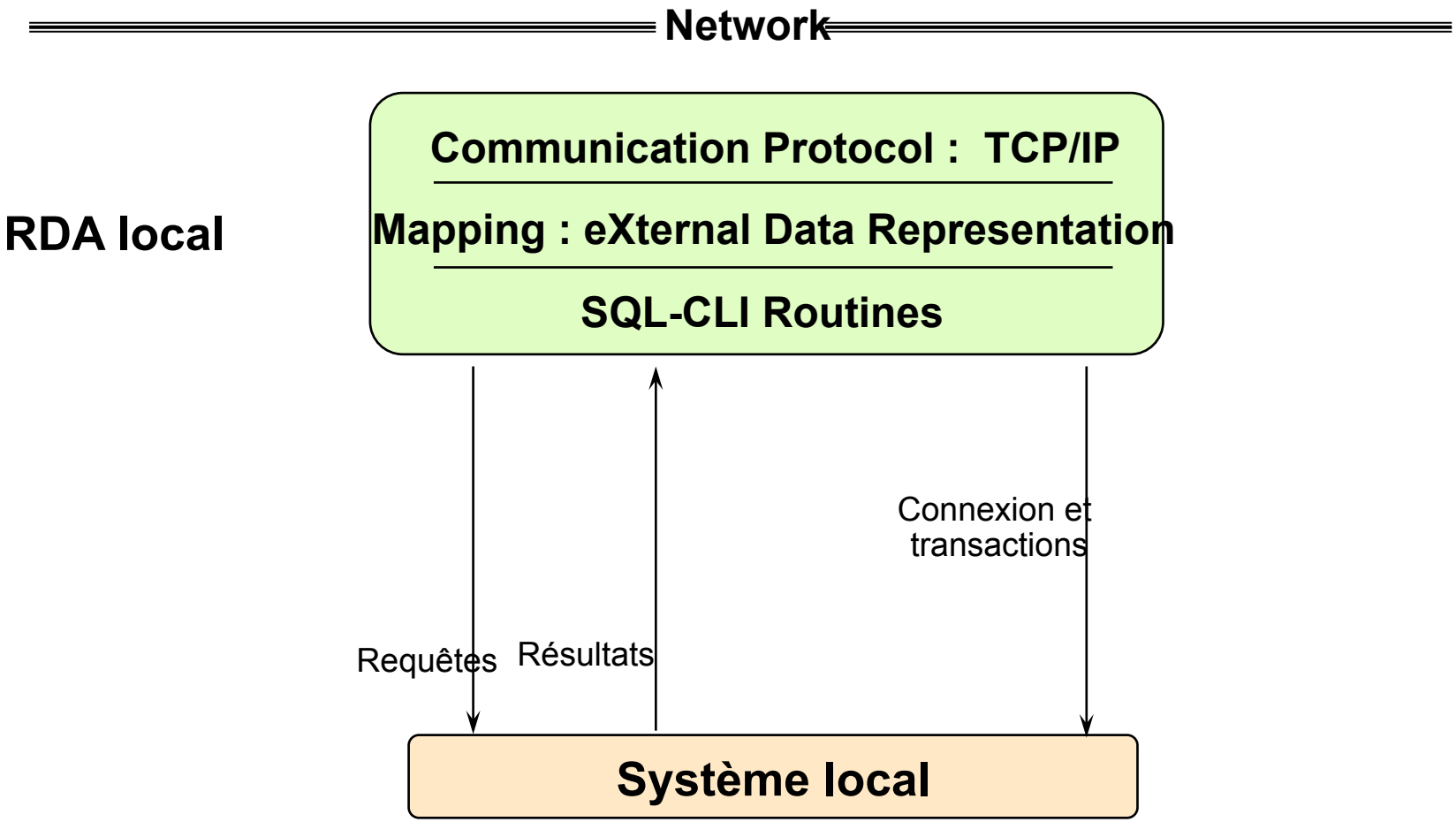
- **Les usagers connaissent la localisation des tables**

- **Si une jointure inter-bases est nécessaire, elle est réalisée par l'application.**

La couche de communication



La couche de communication



Scenario

AllocEnv()

AllocConnect()

Connect()

AllocStmt()

Sending Queries

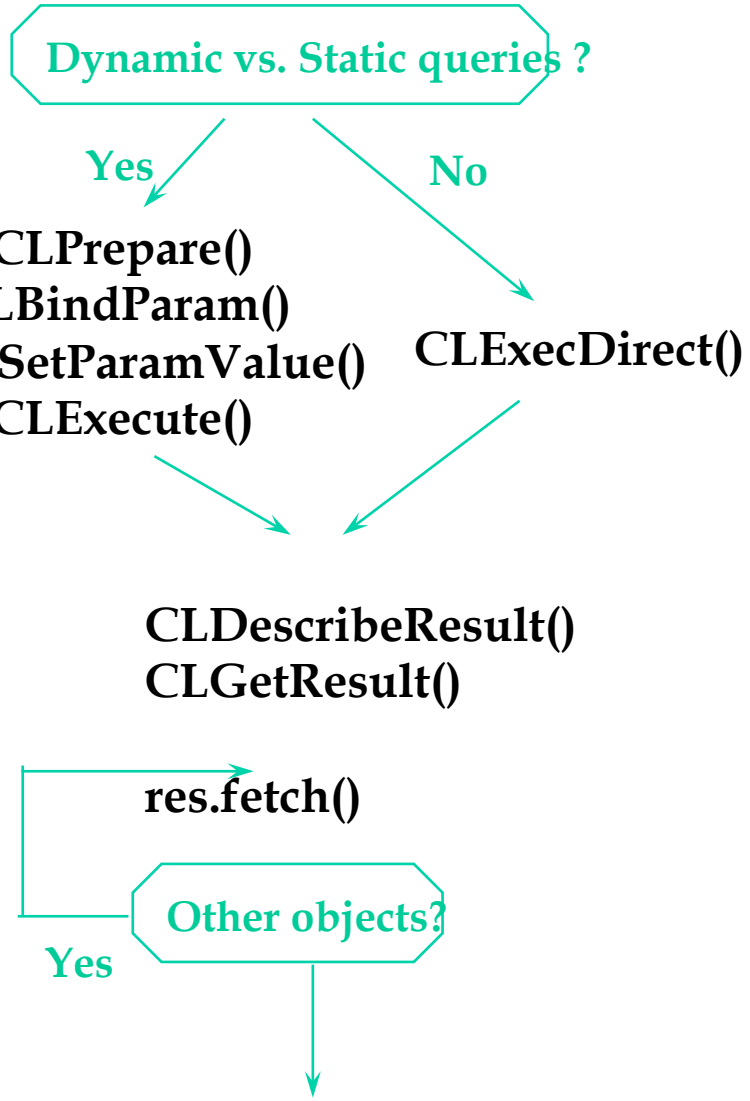
Receiving Results

FreeStmt()

Disconnect()

FreeConnect()

FreeEnv()



Exemple : Sécurité routière

- **Site 1 : base Cartes-Grises**
  - PERSONNE (N°PERS, NOM, PRÉNOM, ADRESSE, …)
  - VOITURE (N°VEH, MARQUE, TYPE, …)
  - CONDUCTEUR (N°PERS, N°VEH, NBACC, …)

- **Site 2 : base SAMU**
  - ACCIDENT (N°ACC, DATE, DEPT, N°VEH, N°PERS, …)
  - BLESSÉ (N°ACC, N°PERS, GRAVITÉ, …)

- **Site 3 : requête**
  - "Liste des blessés graves dans une R18 en région parisienne"

Requête SQL en centralisé

- **Question :**
  - SELECT P.NOM, P.PRENOM
  - FROM PERSONNE P, BLESSÉ B, ACCIDENT A, VOITURE V
  - WHERE P.N°PERS = B.N°PERS
  - AND B.GRAVITÉ > 'Commotions'
  - AND B.NACC = A.NACC
  - AND A.N°VEH = V.N°VEH
  - AND V.MARQUE = 'Renault'
  - AND V.TYPE = 'R18'
  - AND A.DEPT IN (75, 78, 91, 92, 93, 94, 95)

Solution RDA

Q1 / S1 :

```
SELECT    N°VEH
FROM      VOITURE
WHERE     MARQUE = 'Renault'
AND       TYPE = 'R18'
INTO      TEMP1
```

Q2 / S1 :

```
SELECT    *
FROM      PERSONNE
INTO      TEMP2
```

Q3 / S2 :

```
SELECT    B.N°PERS, A.N°VEH
FROM      BLESSÉ B, ACCIDENT A
WHERE     B.GRAVITÉ > 'Commotions'
AND       B.NACC = A.NACC
AND       A.DEPT IN (75, 78, 91, 92, 93,
94, 95)
INTO      TEMP3
```

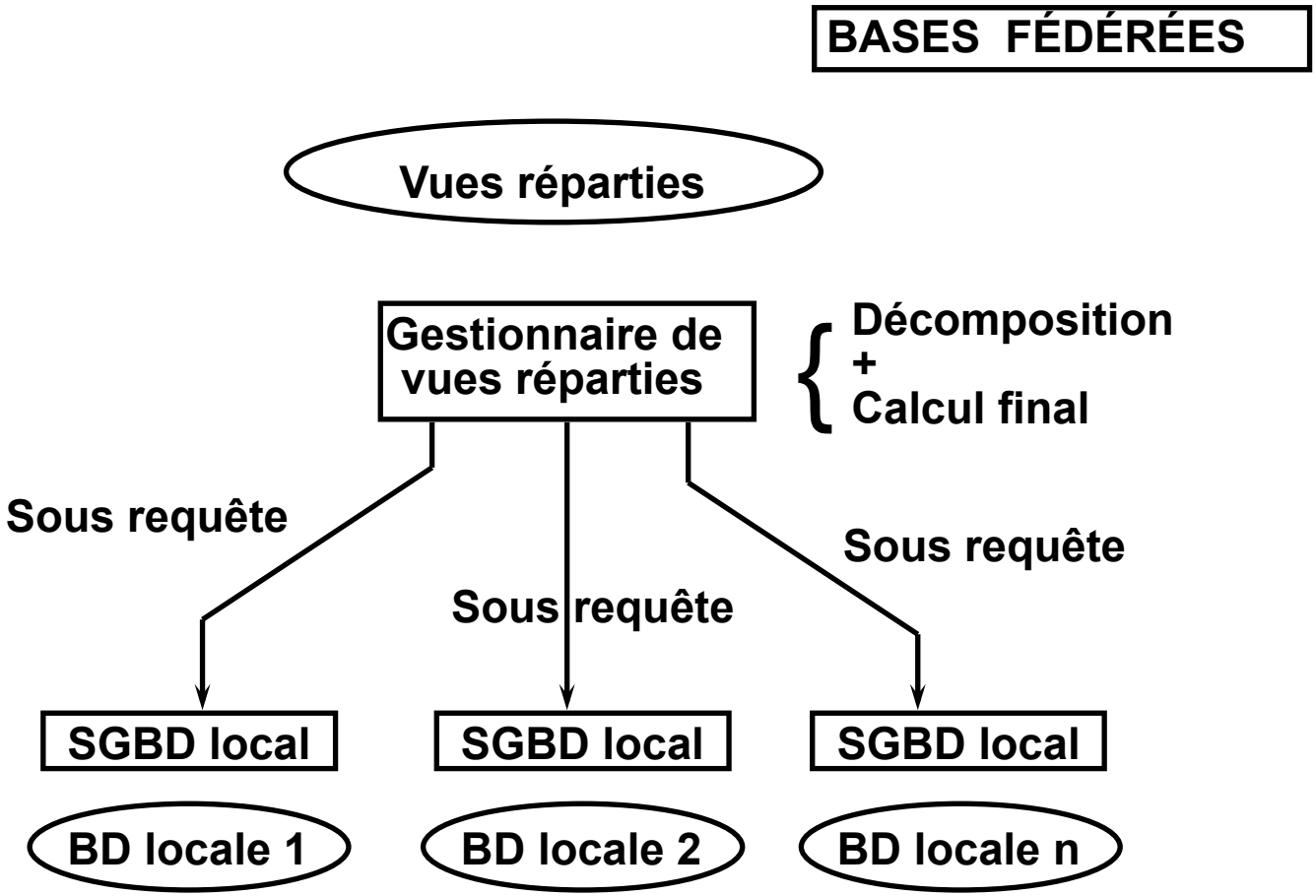
Conclusion RDA

- **Il faut envoyer 3 requêtes différentes pour seulement 2 sites**

- **L'intégration du résultat final doit être fait par l'application :**
  - SELECT P.NOM, P.PRENOM
  - FROM TEMP2 P, TEMP3 B, TEMP1 V
  - WHERE P.N°PERS = B.N°PERS
  - AND B.N°VEH = V.N°VEH

- **La totalité de la relation PERSONNE doit être transférée !**

Niveau 2 : Vues réparties



Vues Réparties : Caractéristiques

- **Un niveau faible de transparence à la localisation est fourni à travers la définition de vues réparties.**
- **Les jointures inter-bases sont exécutées par le système (et non par l'application) .**
- **les mises à jour sont supportées au travers des vues réparties (donc avec des limitations).**

Exemple : vue répartie Accidenté-Grave

- **Site 3 : définition SQL de la vue répartie**

  - CREATE VIEW ACCIDENTÉ-GRAVE
  - (N°PERS, NOM, PRÉNOM, ADRESSE, GRAVITÉ, DEPT, N°VEH, MARQUE, TYPE) AS
    - SELECT P.N°PERS, P.NOM, P.PRÉNOM, P.ADRESSE, B.GRAVITÉ, A.DEPT, V.N°VEH, V.MARQUE, V.TYPE
    - FROM S1.PERSONNE P, S2.BLESSÉ B, S2.ACCIDENT A, S1.VOITURE V
    - WHERE P.N°PERS = B.N°PERS
    - AND B.GRAVITÉ > 'Commotions'
    - AND A.N°VEH = V.N°VEH
    - AND A.N°ACC = B.N°ACC

Requête sur la vue répartie

- **Site 3 : requête**
  - " Liste des blessés graves dans une R18 en région parisienne "

```
SELECT N°PERS, NOM, PRENOM, ADRESSE
FROM   ACCIDENTÉ-GRAVE
WHERE MARQUE = 'Renault'
AND    TYPE = 'R18'
AND    DEPT IN (75, 78, 91, 92, 93, 94, 95)
```

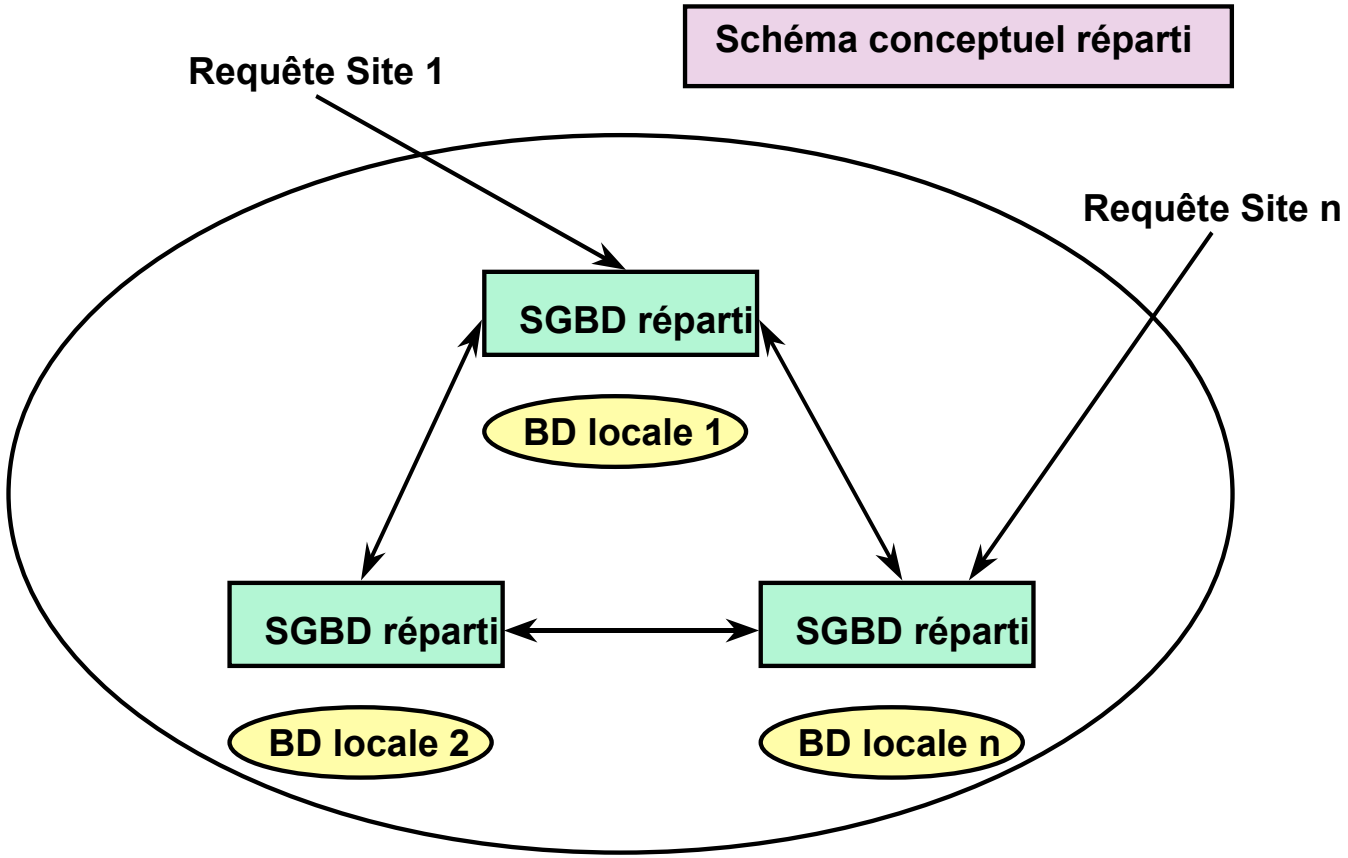
Couche gestion des vues réparties

- **La gestion des vues réparties effectue :**
  - La transformation de la requête sur les relations de base
  - La décomposition de la requête en requêtes mono-sites
    - Q1 / S1 : SELECT N°VEH FROM VOITURE …
    - Q2 / S1 : SELECT * FROM PERSONNE …
    - Q3 / S3 : SELECT B.N°PERS, A.N°VEH FROM BLESSÉ B, ACCIDENT A …
  - Le contrôle de l'exécution des requêtes
  - L'intégration du résultat en effectuant les jointures (et autres opérations) multi-sites

Conclusion Vues Réparties

- **Le système apparaît comme un vrai SGBD réparti**
- **Mais,**
  - Il faut toujours envoyer 3 requêtes différentes pour seulement 2 sites
  - La totalité de la relation PERSONNE doit toujours être transférée !

Niveau 3 : SGBD Réparti



Exemple : schéma réparti

- **Schéma conceptuel :**
  - PERSONNE (N°PERS, NOM, PRÉNOM, ADRESSE, …)
  - VOITURE (N°VEH, MARQUE, TYPE, …)
  - CONDUCTEUR (N°PERS, N°VEH, NBACC, …)
  - ACCIDENT (N°ACC, DATE, DEPT, N°VEH, N°PERS, …)
  - BLESSÉ (N°ACC, N°PERS, GRAVITÉ, …)

- **Implémentation :**
  - Sites 75 à 95 :
    - bases préfectures avec Voiture, Conducteur et Personne pour les voitures immatriculées dans le département
  - Site 2 : base SAMU de la région parisienne (Accident & Blessé)

- **Requête :**
  - "Liste des blessés graves dans une R18 en région parisienne"

Plan d'exécution réparti

- **Q1 / S2 :**
  - SELECT B.N°PERS, A.N°VEH
  - FROM BLESSÉ B, ACCIDENT A
  - WHERE B.GRAVITÉ > 'Commotions'
  - AND B.NACC = A.NACC
  - AND A.DEPT IN (75, 78, 91, 92, 93, 94, 95)
  - INTO TEMP1

  - SEND TEMP1 TO S75, S78, S91, …, S95

Plan d'exécution répartie (2)

- **Q2 / S75, S78, S91, …, S95 :**
  - RECEIVE TEMP1 FROM S2
  - SELECT P.NOM, P.PRENOM
  - FROM PERSONNE P, TEMP1 T, VOITURE V
  - WHERE P.N°PERS = T.N°PERS
  - AND T.N°VEH = V.N°VEH
  - AND V.MARQUE = 'Renault'
  - AND V.TYPE = 'R18'
  - INTO TEMP2.i
  - SEND TEMP2.i TO S3

Plan d'exécution répartie (3)

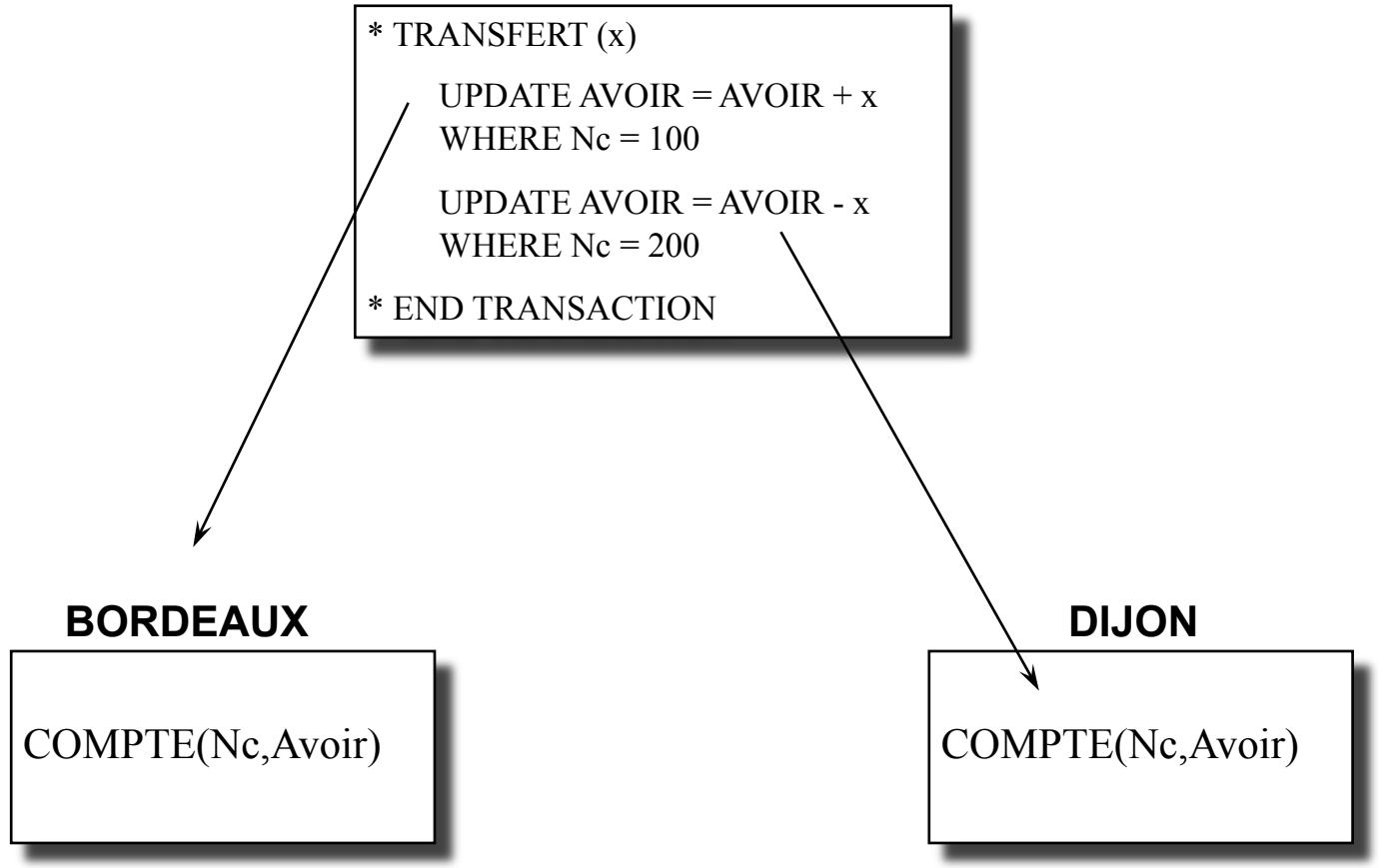
- **Q3 / S3 :**
  - RECEIVE TEMP2.75 FROM S75
  - RECEIVE TEMP2.78 FROM S78
  - …
  - RECEIVE TEMP2.95 FROM S95

  - UNION TEMP2.75, TEMP2.78, …, TEMP2.95
  - INTO RESULT

Conclusion SGBD Répartis

- **La transparence est totale**
- **Les transferts sont minimaux :**
  - Seuls les N° de blessés et des véhicules correspondants sont transférés

- **ATOMICITE :**
  - Garantir qu'un ensemble de mises à jour sur plusieurs sites est totalement exécuté ou pas du tout



Validation à 2 Phases (2PC)

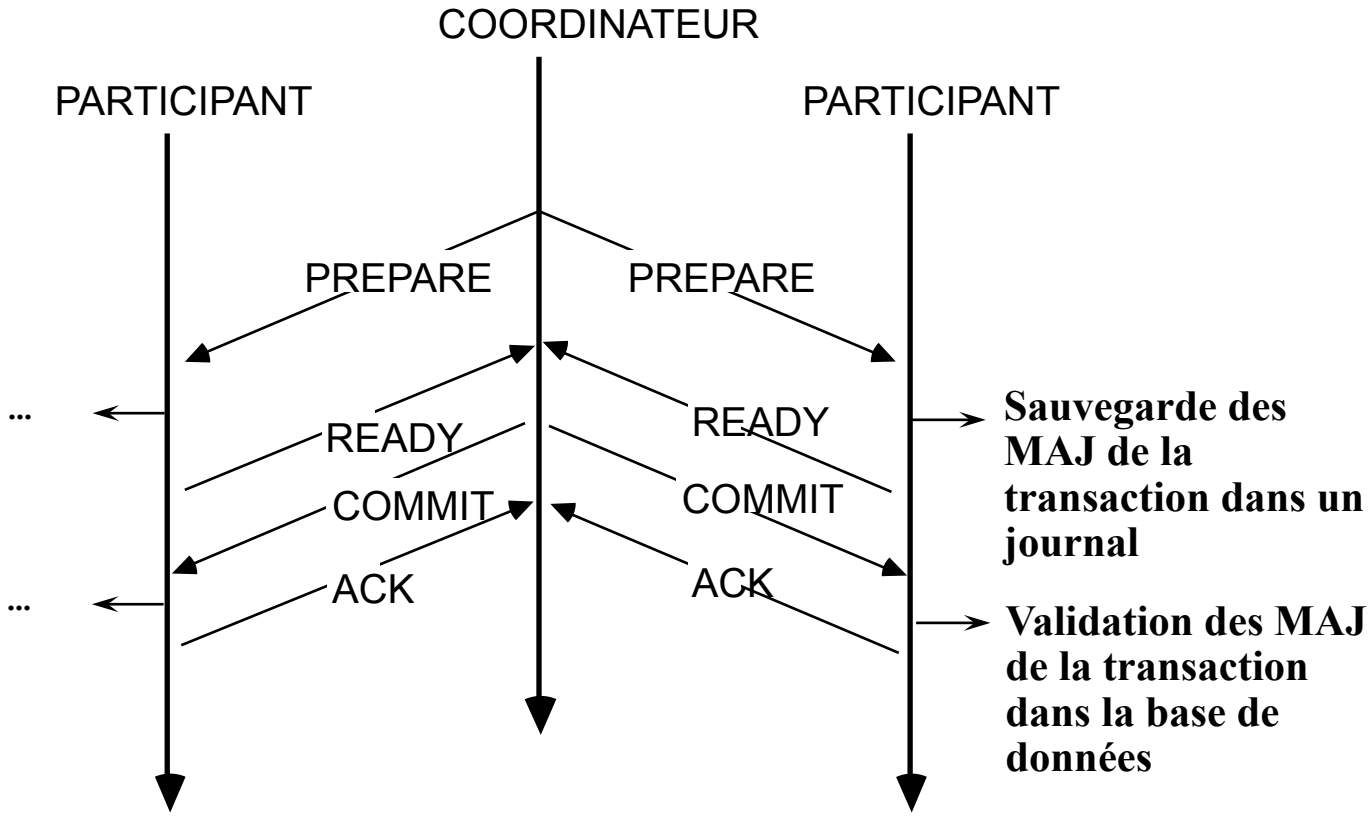
- **Principe : Diviser la commande de validation en deux phases**
  - Phase 1 :
    - Phase de vote pour savoir si tous les sites sont capables de valider ou non la transaction
  - Phase 2 :
    - Phase de décision.
    - La transaction n'est validée que si TOUS les sites votent OK lors de la phase 1.

- **Coordinateur :**
  - Le composant système du site qui centralise et pilote le protocole

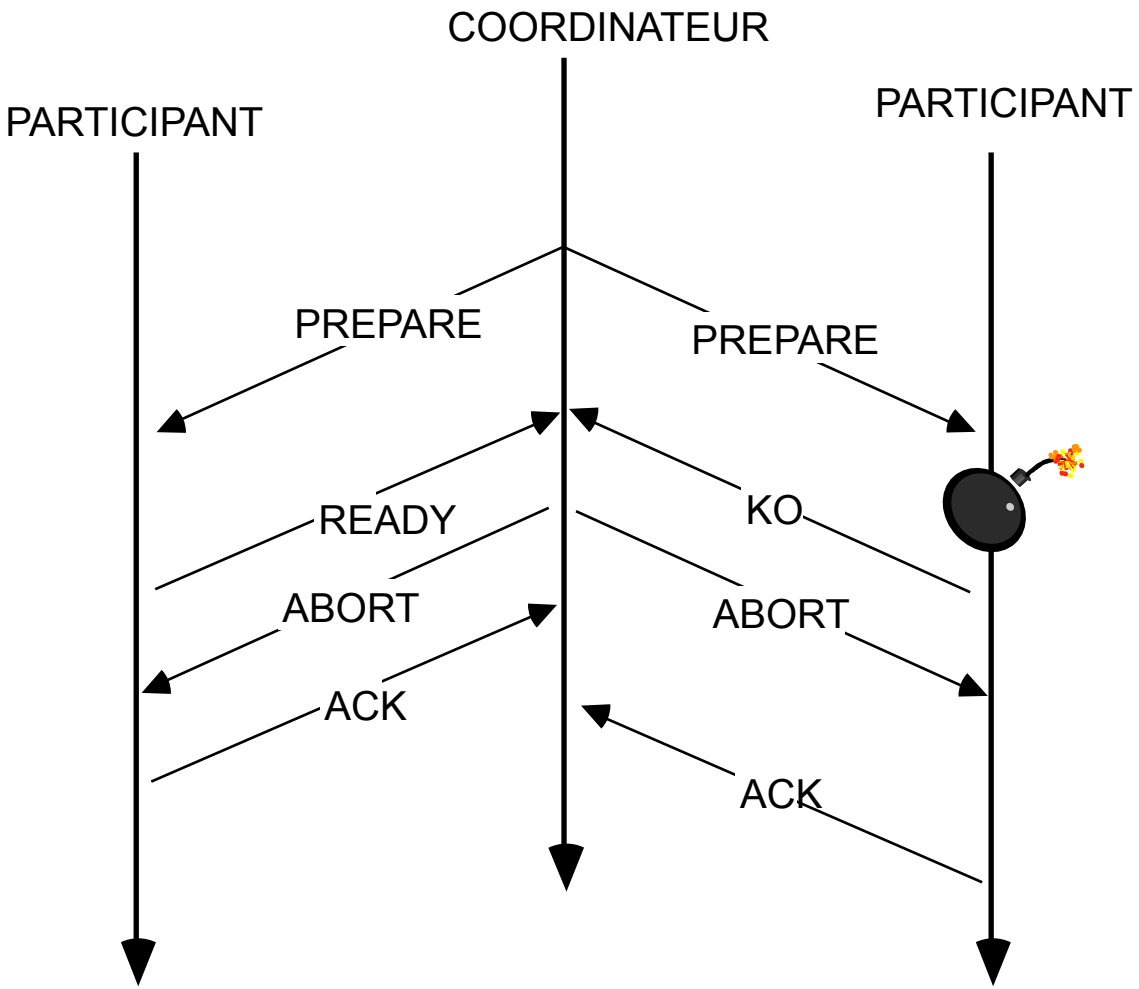
- **Participant :**
  - Le composant système d'un site qui participe à l'exécution de la transaction

- **Ce protocole doit être résistant aux pannes**

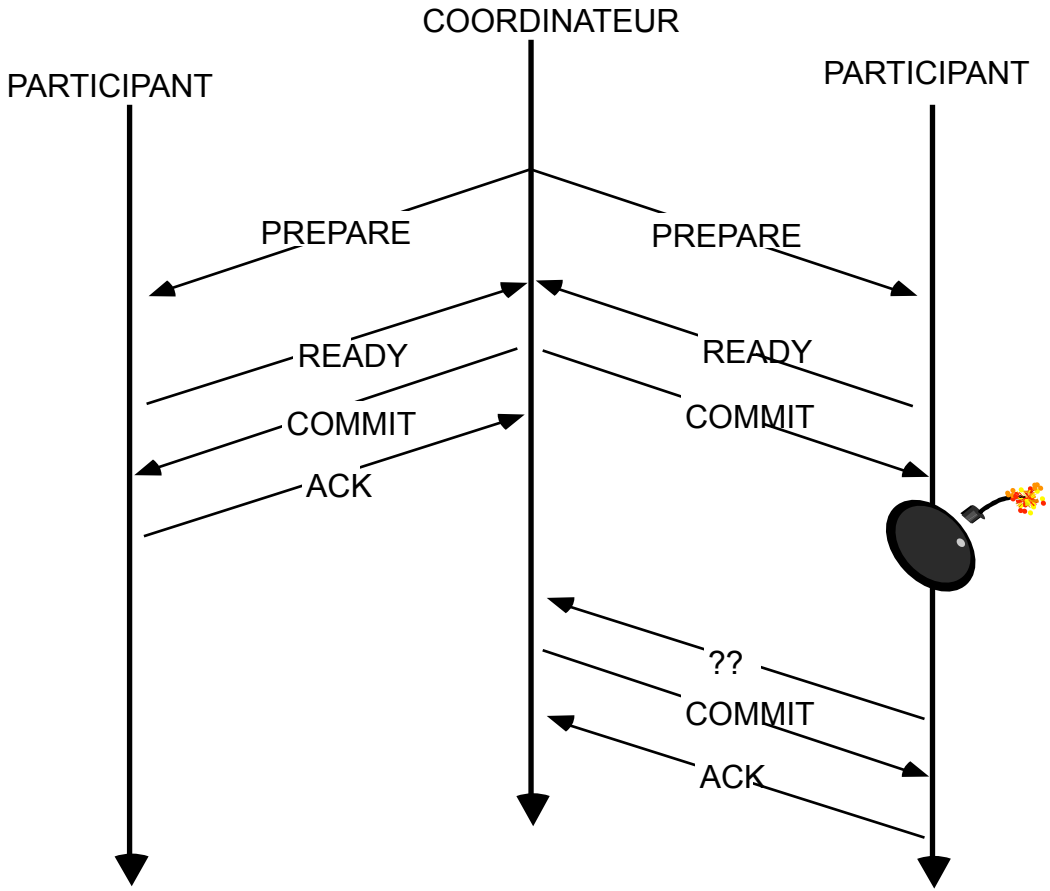
Cas Favorable



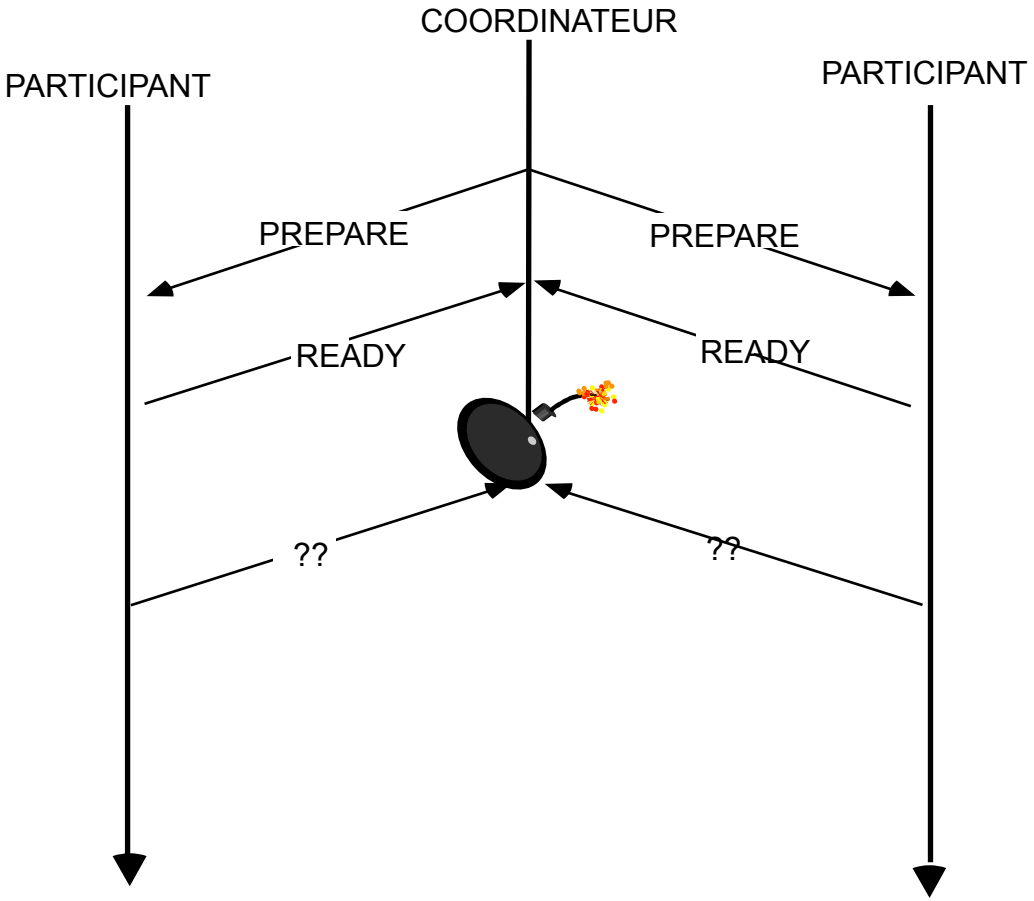
Cas défavorable (1)



Cas défavorable (2)



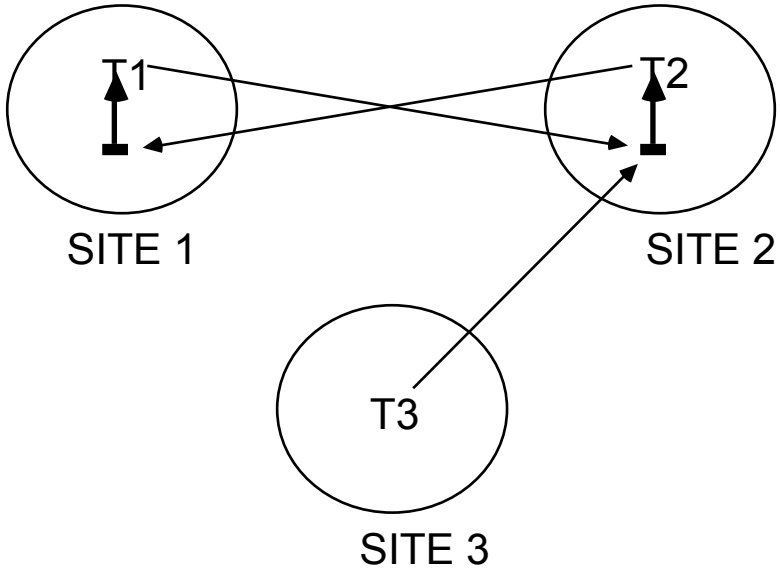
Cas très défavorable (3)



Phase d'incertitude pour le participant:
entre l'envoi du Ready et l'attente de la réponse

Verrou Mortel Distribué

- **La majorité des SGBD sérialisent les transactions grâce à un protocole de verrouillage deux phases**
- **Chaque site est capable de détecter un verrou mortel local**
- **Un contrôle externe est indispensable pour détecter un verrou mortel inter-sites**



Résolution du Verrou Mortel Distribué

- **1) PREVENTION**
  - Garantir que le problème ne survient jamais
  - Combinaison de verrouillage et d'estampillage (DieWait, WoundWait)

- **2) DETECTION**
  - Construction d'un graphe d'attente global par union des graphes d'attente locaux
  - En cas de cycle, abandon d'une des transactions du cycle

- **3) PRESOMPTION**
  - Annulation des transactions n'ayant pas terminé leur exécution après un certain délai (timeout)

Gestion de Copies multiples

- **Objectif :**
  - Assurer la gestion en temps réel ou en différé de copies multiples de la même données afin d'en augmenter la disponibilité

- **Avantages :**
  - en cas de panne d ’un site les données restent disponibles sur un autre site

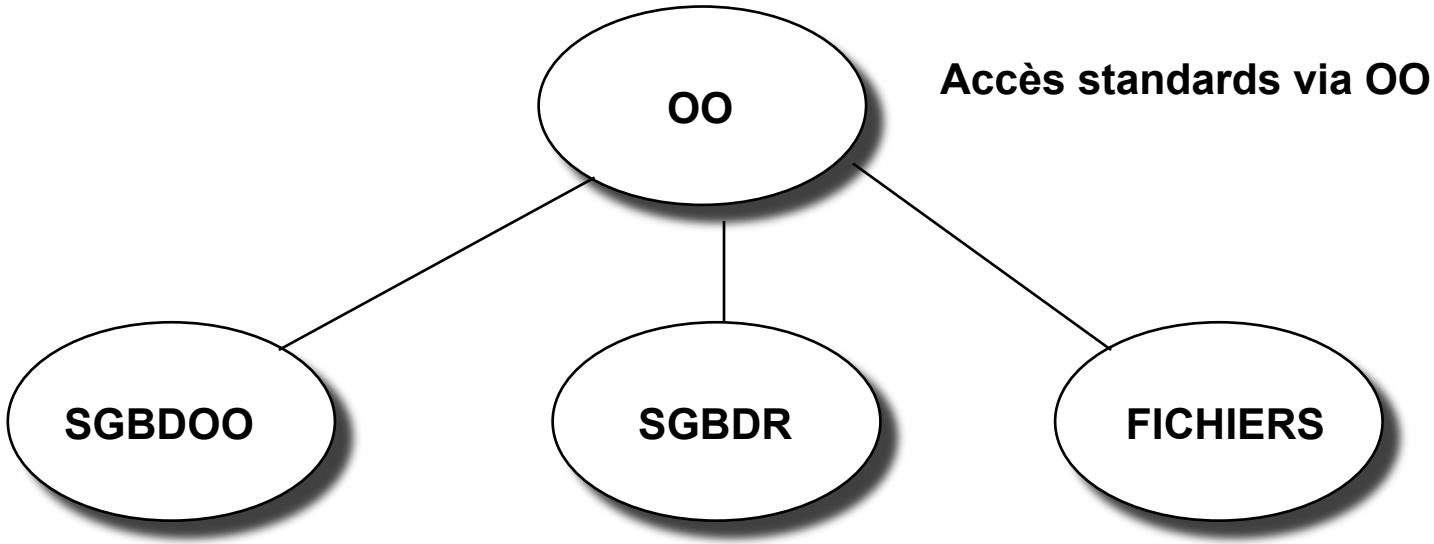
- **Attention :**
  - nécessité de mises à jour sur tous les sites
  - problèmes de convergences (garder les copies identiques)

- **Le vote majoritaire :**
  - garantit qu ’une majorité de copiés reçoit chaque mise à jour
  - une transaction ne peut être commise que si une majorité de copies vote « PRÊT »

Bases de Données Hétérogènes

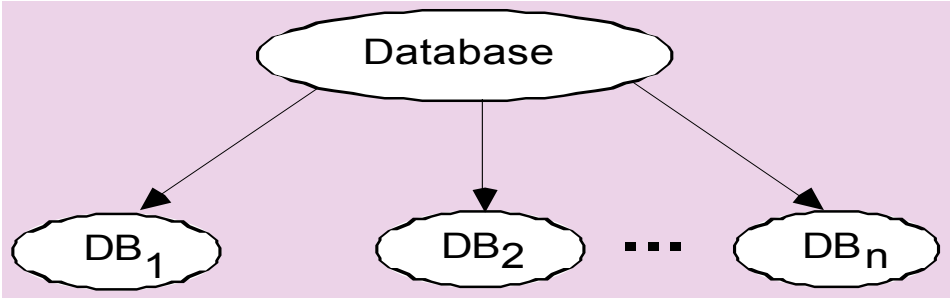
- **OBJECTIF :**
  - Permettre la gestion de BD réparties par des SGBD différents

- **Avantage :**
  - accéder des bases existantes gérées par d ’autres SGBDs à l ’aide d ’un langage uniforme.

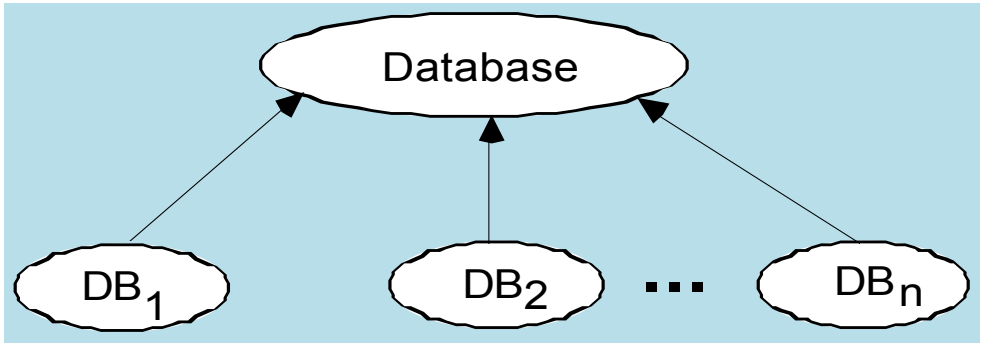


Fédération de bases locales

- **BD répartie : division de la BD en BD locales**



- **BD fédérée : intégration de BD locales existantes**



Problèmes de l'Hétérogénéité

- **Uniformisation des schémas locaux**
- **Intégration en un schéma multi-base**
- **Traduction des requêtes en accès locaux**
- **Optimisation selon les capacités des systèmes participants**
- **Coopération des contrôles (concurrence, atomicité, …)**

CONCLUSION

- **Le relationnel offre un cadre homogène pour le réparti :**
  - Une représentation naturelle des données
  - Un langage universel standardisé (SQL)
  - Des SGBD modernes et flexibles
- **Le réparti homogène existe :**
  - Atteinte partielle des objectifs
  - Généralisation des solutions client-serveur
- **Le réparti hétérogène reste difficile**