My experience working to consolidate, manage, coordinate multiple databases so they talk to each other.

DB-Link

As a DBA in Oracle inc. It was fairly common for me to create db connections using DB-link between databases and access the relations using relation@dblink command. I ran customer POC and had to mimic customer environments for ISPs. The rate-plans were stored in a product side database. The sales-person for a particular type of customer (regular customer, corporate customer, business customer etc) had access to only their type of rate-plan tiers . Hence a DBlink was created from sales database to product database to access the rate plans. I made changes to listener.ora and tnsnames.ora before using sql to configure the dblink. Some of the customers had an existing MS-Sql server solution for the product database. For such environments I used the Linked Servers feature on SQL Server.

HA- High Availability

In almost all my previous jobs (Oracle, Yahoo, Salesforce) I have set up HA several times.
In Postgres I have setups 1 Master and 2 warm-standby, with each standby having its storage for shared-nothing architecture. Only Master Database has the permission to write. using Write-Ahead-Log (WAL) the change log is shipped to the other 2 standby. These changes are applied to every standby to keep the databases in sync. An HA- heartbeat process checks the health of the master database. If it fails configured-number of conductive times a failover is initiated. Both the standby database try to stake their claim. The DB that is picked becomes the master database and the other remains the standby. DBA should pay attention to these failovers and configure another standby for fully functional HA otherwise we are one failure away from a business-stop scenario. Depending on the type of failure, most checkpointed transactions are reflected in the standby database. Any in-flight transactions are lost unless handled at the application level.
In Oracle I have setup RAC HA setup
In MS-SQL server I have set up a db mirror solution.

MPP - Massively Parallel Processing

I have worked on Greenplum database a product for Big-Data created by Greenplum renamed to Pivotal Software
This is a shared nothing architecture with one master database and multiple compute (or segment) nodes each having their segment of data. There is a compute node list maintained by the master. The master database acts as a coordinator and consolidator. Once it receives the SQL for processing, the sql gets sent to all the compute nodes who process the sql on their own segment of data parallelly (hence Parallel processing). This result is sent back to the master database who then consolidates the result and gives it back to the user.
For availability an HA may be configured for Master and each of the segment databases.