



CompleteSchool Database Organization

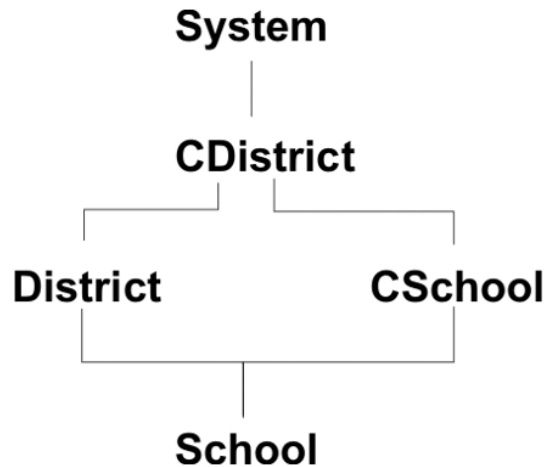
In CompleteSchool database documentation, you will find references to “*data sources*”. A data source represents a class of databases that all have identical definitions (but different table contents). For example, the **DsSchool** data source is used to access school information, but the school, district, and fiscal year are combined to determine the actual database name. For example, the **DsSchool** data source might refer to the **DsSchool_S351_D720_Y2000_2001** database.

In CompleteSchool, data sources are logical databases. That is, CompleteSchool programs translate data source names into actual database names. User settings tell which databases to use.

The following data sources are used:

- DsSystem** System-wide information (e.g. user, location, database, security, and other information used commonly). There is only one such database system-wide. The database name is the same as the data source name.
- DsServer** Server specific information including information specific to the users that access the server. There is one such database per server. The database name is the same as the data source name.
- DsCDistrict** Continuous (usually permanent) District-wide information. There is one such database per District. This information contains district specific code tables and date tagged information as well as student information that does not change when a student moves between schools. Some tables have school and fiscal year information to identify where and when the information originated. For example, student enrollment history is stored here, but it originates from enrollment transactions at a specific school in a specific fiscal year. The data in this database include student grade and test history, and this database usually has about as much information as all the other databases put together.
- DsCSchool** Continuous (usually permanent) School information. There is one such database per school. For the most part, this is information that applies to the school and student body as a whole (e.g. graduation requirements, plans, locker information, etc.).
- DsDistrict** Year based District-wide information. That is, information used throughout the District that is likely to change somewhat from year to year (e.g. courses offered, schools in the district, etc.). There is one such database per District per year.
- DsSchool** School information for a single school year (or session). There is one such database per School per year. This is mostly student information that is either frequently accessed or is of little value outside the specific school and year (e.g. student demographics, attendance, class assignments, locker assignments, etc.).
- DsPersonnel** Personnel information. There is one such database per District. Most employee information is stored here.

The database organization is roughly hierarchical as follows:



Databases have views that access the tables stored in databases above them in the hierarchy. Databases have some views that reference the tables farther down the tree. Databases give context to procedures running within those databases.

This database organization has many advantages and some disadvantages:

Advantages:

High performance. School based operations for schools are always performed using a small data set with limited impact on other schools. For example, period by period attendance updates account for a large amount of the computer time spent in most high schools. Keeping these tables small increases performance, and locks on attendance data affect only the attendance data for one school.

High Scalability. Databases are easily organized across multiple servers. Typical school based operations (e.g. attendance updates) affect that school only.

Easier, context based programming. Programs are easier to understand and more likely to optimize properly because every access doesn't have to reference the proper school, year, and district.

Database organization more closely follows the security model.

Old year databases can be totally write protected to prevent accidental modifications.

Disadvantages:

Queries across multiple databases take slightly longer than queries to a single database.

Software updates take longer when there are more databases to update.