25-point Basic MySQL Setup/Optimization Checklist

©2009 Jamie Estep · www.saynotoflash.com

	Server Setup	**Complete**	
1.	Root user is setup with a secure password, and local access only.	✓	✗
2.	Have a backup and restoration plan in place, in case database server is completely destroyed.	✓	✗
3.	Install method or program to benchmark performance, and enable the slow query log.	✓	✗
4.	Turn off MySQL's DNS service, and authenticate by IP only if possible.	✓	✗
5.	Only grant access for specific "privileges ON db TO user @ host". Do not ever use: GRANT ALL ON EVERYTHING TO USER@ANYWHERE.	✓	✗

	Schema Design		
6.	Name database, tables and column with lower case letters and an "_" separating words. `id` is not a descriptive name!	✓	✗
7.	Database, tables, and columns are using the same collation and character set and not UTF-8 unless required.	✓	✗
8.	Foreign keys being used to maintain data integrity - N/A for MyISAM.	✓	✗
9.	Tables segmented logically based on their purpose.	✓	✗
10.	Don't use any reserved word *(date, int, decimal, time, etc.)* for a database, table or column names.	✓	✗

	Table Design		
11.	Use proper data types. For example, numbers and dates not stored in character or text fields.	✓	✗
12.	Use unsigned int(10) instead of bigint for columns requiring large numerical keys.	✓	✗
13.	Use the smallest length for the data type being used. No VARCHAR(255), BIGINT(20), DECIMAL(20,2), etc.	✓	✗
14.	Avoid using TEXT and BLOB data types whenever possible.	✓	✗
15.	Do not store data in a non-relational format *(value1,value2,value3, in a single column)* . Use multiple tables and joins to retrieve one-to-many relationships and to preserve data integrity.	✓	✗

	Index Optimization		
16.	Use the proper indexes *(unique, normal, partial, etc.)* for the data, and do not create duplicate indexes.	✓	✗
17.	Don't modify an indexed field in a query or the index will not be used. ~~" WHERE DATE(`date_of_birth`) > '2009-07-01' "~~	✓	✗
18.	Use multi-column *(concatenated)* indexes for columns that frequently get queried together.	✓	✗
19.	Try to use indexes in columns with a high cardinality *(many unique values in relation to the total rows)* .	✓	✗
20.	Avoid indexing columns with very few unique values or many null's. An empty value is better than NULL.	✓	✗

	Query Optimization		
21.	Use specific column names when selecting. Avoid: SELECT *.	✓	✗
22.	Use MySQL's built-in functions *(COUNT, SUM, AVG, etc.)* instead of application level functions.	✓	✗
23.	Do not select TEXT or BLOB columns unless absolutely necessary.	✓	✗
24.	Use transactions whenever applicable.	✓	✗
25.	Use SQL_NO_CACHE for very large queries, and queries on data that changes frequently.	✓	✗

	Bonus - 5 Tips and Tricks		
1.	Use TIMESTAMP and not DATETIME for all date fields newer than Jan 01 1970.	✓	✗
2.	Never use SIGNED INT fields unless you actually need to store negative numbers.	✓	✗
3.	The _ci at the end of a collation means "case insensitive". Use _ca collation if you care about case sensitivity.	✓	✗
4.	Know that InnoDB is often a better choice than MyISAM. Don't use MyISAM just because it is default and somebody told you it's better.	✓	✗
5.	If you're making a serious application and you are unsure of your database design and management ability, consult a qualified MySQL expert to prevent major problems down the road. Spending some money now could save your business down the road.	✓	✗