Handout

Database Manipulation API

If you are having problems with your database queries whilst developing, you can enable additional debugging information by calling the `$DB->set_debug()` method, passing a boolean `true` parameter.

In the following table, the parameters quoted translate to the following:

- `$table`: the database table without the prefix
- `$conditions`: the multidimensional array mentioned earlier
- `$select`: the WHERE clause string
- `$sql`: an SQL command with brace-enclosed table names
- `$params`: the placeholder values array
- `$field/$fields`: $field name(s)
- `$sort`: a comma separated field list or ‘`*`’
- `$limitfrom`: limit start number
- `$limitnum`: limit number of records
- `$strictness`: strictness constant as discussed earlier
- `$countitem`: the count string to be used in the SQL call – default is COUNT('x')
- `$newvalue`: the new value for the field
- `$dataobject/$dataobjects`: standard class objects representing a table record – the keys mirror the field names
- `$bulk`: set to true if further operations can be expected – defaults to `false`
- `$returnid`: defaults to `true` – should the record’s id be returned?

Getting a single record	
	get_record() **PARAMS**: `$table, $conditions, $fields, $strictness`
	get_record_select() **PARAMS**: `$table, $select, $params, $fields, $strictness`
	get_record_sql() **PARAMS**: `$sql, $params, $strictness`
Getting multiple records	
	get_records() **PARAMS**: `$table, $conditions, $sort, $fields, $limitfrom, $limitnum`
	get_records_select() **PARAMS**: `$table, $select, $params, $sort, $fields,`

	`$limitfrom, $limitnum`
	get_records_sql() **PARAMS**: `$sql, $params, $limitfrom, $limitnum`
	get_records_list() **PARAMS**: `$table, $field, $values, $sort, $fields, $limitfrom, $limitnum`
Getting data as key/value pairs in an associative array	
	get_records_menu() **PARAMS**: `$table, $conditions, $sort, $fields, $limitfrom, $limitnum`
	get_records_select_menu() **PARAMS**: `$table, $select, $params, $sort, $fields, $limitfrom, $limitnum`
	get_records_sql_menu() **PARAMS**: `$sql, $params, $limitfrom, $limitnum`
Counting records that match the given criteria	
	count_records() **PARAMS**: `$table, $conditions`
	count_records_select() **PARAMS**: `$table, $select, $params, $countitem`
	count_records_sql() **PARAMS**: `$sql, $params`
Checking if a given record exists	
	record_exists() **PARAMS**: `$table, $conditions`
	record_exists_select() **PARAMS**: `$table, $select, $params`
	record_exists_sql() **PARAMS**: `$sql, $params`
Getting a particular field value from one record	
	get_field() **PARAMS**: `$table, $field, $conditions, $strictness`
	get_field_select() **PARAMS**: `$table, $return, $select, $params, $strictness`
	get_field_sql() **PARAMS**: `$sql, $params, $strictness`
Getting field values from multiple records	
	get_fieldset_select() **PARAMS**: `$table, $return, $select, $params`
	get_fieldset_sql() **PARAMS**: `$sql, $params`
Setting a field value	

	set_field() **PARAMS**: `$table, $field, $newvalue, $conditions`
	set_field_select() **PARAMS**: `$table, $newfield, $newvalue, $select, $params`
Deleting records	
	delete_records() **PARAMS**: `$table, $conditions`
	delete_records_select() **PARAMS**: `$table, $select, $params`
Inserting records	
	insert_record() **PARAMS**: `$table, $dataobject, $returnid, $bulk`
	insert_records() **PARAMS**: `$table, $dataobjects`
Updating records	
	update_record() **PARAMS**: `$table, $dataobject, $bulk`
Using record sets	
	get_recordset() **PARAMS**: `$table, $conditions, $sort, $fields, $limitfrom, $limitnum`
	get_recordset_select() **PARAMS**: `$table, $select, $params, $sort, $fields, $limitfrom, $limitnum`
	get_recordset_sql() **PARAMS**: `$sql, $params, $limitfrom, $limitnum`
	get_recordset_list() **PARAMS**: `$table, $field, $values, $sort, $fields, $limitfrom, $limitnum`

Cross-DB Compatibility

The following `$DB` methods ensure that SQL statements are compatible between the supported databases. Please view the Moodle documentation’s page for examples of the use of each of the functions. They are mentioned here to make you aware that they exist.

Function	***Notes***
get_in_or_equal	Constructs 'IN()' or '=' SQL fragment and returns an SQL snippet and a parameter array to specify if a value is IN the given list of items.
sql_bitand	Returns snippet to be used to perform bitwise operations.
sql_bitnot	
sql_bitor	

sql_bitxor	
sql_cast_char2int	Returns the SQL to be used to CAST one CHAR column to INTEGER or a REAL number. Ensure the CHAR column you're trying to cast contains real numbers or the database will throw an error!
sql_cast_char2real	
sql_ceil	Returns the cross-DB correct CEIL (ceiling) SQL expression applied to the field name. Note CEIL($fldname) is the default.
sql_compare_text	Returns the snippet to be used to compare one TEXT (clob) column with a VARCHAR column, because some databases don't support this type of comparison.
sql_concat	Returns a snippet to do CONCAT between the field names passed and with *sql_concat_join()*, using passed in character(s) as the separator.
sql_concat_join	
sql_equal	Returns an equal (=) or not equal (<>) snippet. Caution advised.
sql_fullname	Returns the proper snippet to concatenate user’s first name and last name as a full name.
sql_intersect	Returns the snippet to find the intersection of two or more queries.
sql_isempty	Returns the snippet to query whether one field is empty or not.
sql_isnotempty	
sql_length	Returns the snippet to be used to calculate the length of characters of the field.
sql_like	Returns 'LIKE' snippet of a query and/or escape the LIKE special characters such as '_' or '%'.
sql_like_escape	
sql_modulo	Returns the snippet to be used to perform module '%' operation – remainder after division.
sql_null_from_clause	Returns an empty FROM clause.
sql_order_by_text	Returns the snippet to be used to order by one TEXT (clob) column. Caution recommended.
sql_position	Returns the snippet for searching one string with the location of another.
sql_regex	Returns the driver-specific snippet syntax for matching regex.
sql_regex_supported	Checks if this database driver supports regex syntax when searching.
sql_substr	Returns the proper snippet used to extract substrings.