# User Guide

## 1. SQL Syntax

We have implemented a subset of the SQL language which emulates its behaviour as closely as possible. In the following sections of this short user guide, we have supplemented the relevant sections with SQL format and examples. Take note of the following notations in the following sections:

* Angle brackets [ ] are segments of SQL syntax which are optional.
* Arrow brackets < > are not part of the SQL syntax
* Words in upper case are reserved keywords of SQL, for e.g. SELECT, WHERE, etc.
* Words in lower case may refer to column names, values, view names, or table names.
* Every SQL query has to end with a semi-colon ;
* Values are represented as integer constants, e.g. 100, 128, etc.
* Strings are enclosed with single-quotations, e.g. ‘Chan Mali Chan’, ‘Tan Ah Kow’, etc.

## 4. Defining Views

In this project, we have implemented projections, restrictions and PK-PK inner joins on database views. The database administrator may use any combinations of such to build his views of interest. White spaces are not allowed when defining names of views.

### 4.1 Projection and/or Restriction

To define views on a single physical table, enter the SQL command according to the syntax as follows:

CREATE VIEW view\_name AS SELECT <projection> FROM physical\_table

[WHERE <restriction>];

If there is no projection required, then replace <projection> with the asterisk \* to refer to all columns. To define a projection to the view, separate each column name with a comma , . For example, emp\_id, name, address, salary.

The optional WHERE clause is used to extract only tuples that fulfill a certain criterion, i.e. the restriction of the view. The following operators can be used in restriction:

|  |  |
| --- | --- |
| Operator | Description |
| = | Equal |
| != | Not equal |
| > | Greater than |
| >= | Greater than or equal |
| < | Less than |
| <= | Less than or equal |
| AND | True only if both the first and second condition is true. Else false. |
| OR | True if either the first or the second condition is true. Else false. |

Additionally, the following arithmetic operators can be used in restriction:

|  |  |
| --- | --- |
| Operator | Description |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulo (Remainder of division) |

### 4.2 PK-PK Inner Join

To define views for PK-PK inner-joins, enter the SQL command according to the syntax as follows:

CREATE VIEW view\_name AS SELECT <projection> FROM phyTable1

INNER JOIN phyTable2

ON phyTable1.id = phyTable2.id

[WHERE <restriction>];

The ON clause specifies the predicate for the inner join. The view will join physical table 1 phyTable1 and physical table 2 phyTable2 using the primary key id of both tables. Restriction and projection may also be defined on join views. When specifying projection for PK-PK inner-joins, table name must be prepended to the column name. For example, table1.column1, table1.column2, table2.column1.

## 5. Dropping Views

You can drop a view using the following SQL command:

DROP VIEW view\_name;

View will be removed from the database.

## 6. Querying Views

To query a particular view, enter the SQL command according to the syntax as follows:

SELECT <column\_name(s)> FROM view\_name [WHERE <restriction>];

Column names are seperated by comma , . For example, column1, column2, column3… Column names can also be replaced by asterisk \* to indicate the selection of all columns. Restriction may be defined when querying views.

## 6.1. Insertions

To insert a tuple into the view, enter the SQL command according to the syntax as follows:

INSERT INTO view\_name (<column1, column2, column3,…>)

VALUES (<value1, value2, value3,…>)

A tuple will be inserted based on the mapping between column name(s) and the corresponding value(s). For example, value1 will be inserted to column1 and value2 will be inserted to column2 for the particular tuple.

## 6.2 Deletions

To delete tuple(s) from the view, enter the SQL command according to the syntax as follows:

DELETE FROM view\_name [WHERE <restriction>];

Restriction may be defined on deletions. If no restriction is indicated, all tuple in the view will be deleted.

## 6.3. Updates

To update tuple(s) in the view, enter the SQL command according to the syntax as follows:

UPDATE view\_name SET <column1=value1, column2=value2,…>

[WHERE <restriction>];

Tuples matching the restriction criterion will have the corresponding value that maps to the column name updated. For instance, the value of column1 of a particular tuple will be updated to value1. Restriction (WHERE clause) may be defined on a update query. If no restriction is indicated, all tuples in the view will have their columns updated to their corresponding values. For instance, value of column1 will be updated to value1.