SQL Notes:

1. Retrieving data from the database:

The basic form of a select statement is:

SELECT [options] items

[INTO file\_details]

FROM tables

[ WHERE conditions ]

[ GROUP BY group\_type ]

[ HAVING where\_definition ]

[ ORDER BY order\_type ]

[LIMIT limit\_criteria ]

[PROCEDURE proc\_name(arguments)]

[lock\_options];

1. Users can use the DESCRIBE <tablename> or DESC <tablename>

statement to see the structure of the table.

1. SELECT \* FROM CAT; will return TABLE\_NAME & TABLE\_TYPE.
2. SQL is used to interact with a database to manage and retrieve data.

SQL is used to retrieve data from the database. The DBMS processes the SQL request, retrieves the requested data from the database, and returns it. This process of requesting data from the database and receiving back the results is called a database query and hence the name Structured Query Language.

Data Types: The description of the kinds of data stored, passed and used.

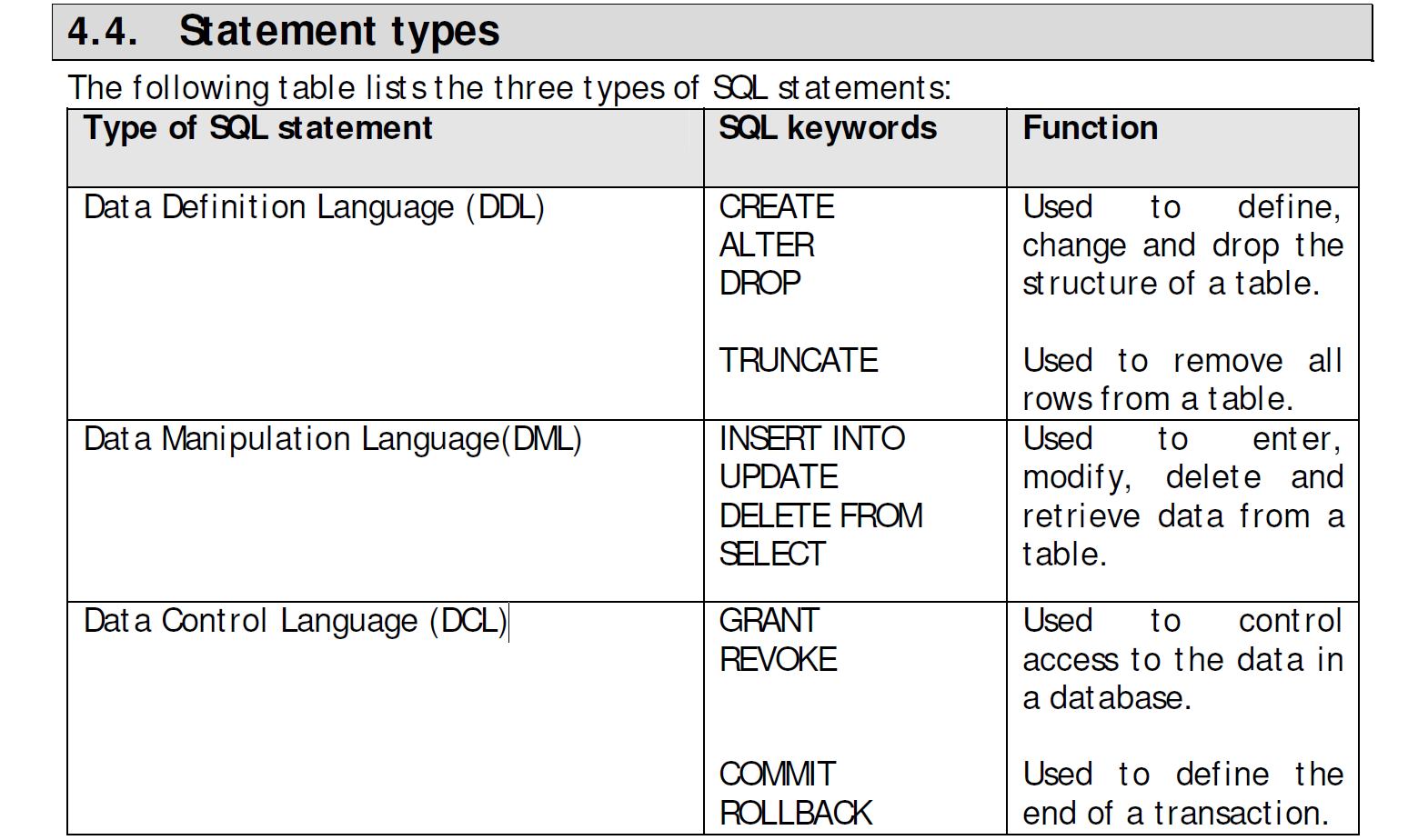
Backward Compatible: A design that continues to work with earlier versions of a language, program, etc.

1. Index

Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries.

1. Table constraints must be used when constraint is applied for more than one column of a table.
2. A UNIQUE constraint on a column(s) means that the values in the column(s) should be distinct although it can have NULL values.
3. The NULL value occupies space.
4. SQL Statement:

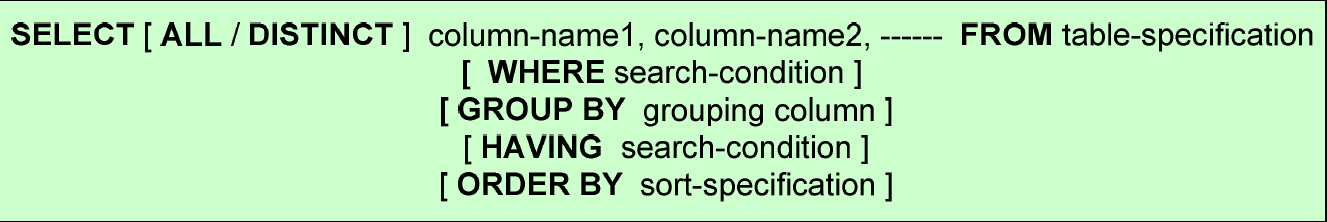
* DESCRIBE Customer\_Details; or DESC Customer\_Details;



1. SELECT Statement

The SELECT statement retrieves data from a database and returns it in the form of query results. The result of a SQL query is always a table of data.

SELECT statement syntax diagram:



1. It is a good programming practice to avoid the use of SELECT \*. It is better to list the column names.
2. WHERE Clause

The WHERE clause is used to specify a search condition that limits the number of rows retrieved. It is a row wise operation.

For each row, the search condition can produce one of the three results:

· If the search condition is t rue, the row is included in the query results

· If the search condition is false, the row is excluded from the query results

· If the column being searched has a NULL value, the row is excluded from the query results

If the list of conditions is not known at compile time and is instead built at run time, you don't have to worry about whether you have one or more than one condition. You can generate them all like:

and <condition>

and concatenate them all together. With the 1=1 at the start, the initial and has something to associate with.

I've never seen this used for any kind of injection protection, as you say it doesn't seem like it would help much. I have seen it used as an implementation convenience. The SQL query engine will end up ignoring the 1=1 so it should have no performance impact.