# Symbols

{ z | r } – choose between one of these, but must pick one and only one.

[x] – “x” is optional

[x | y | z] – “x” or “y” or “z” can be entered here, but they are optional. Only one of these options can be entered.

<banana> - “banana” is defined somewhere else in the document

<banana> ::== - this is the definition of “banana” and should replace all instances of “<banana>”

# Written

CREATE (all words that are uppercase) – write exactly “create” or whatever the keyword is

replace\_me\_with\_stuff (lowercase with underscores all one word) – repace this with whatever belongs here (ex. “column\_name” would be changed to the name of the column you are targeting)

[ ,...n ] – write the thing before this as many times as you want, separate by a comma

All of Claire’s notes are green and should not be taken literally as syntax

# Examples

|  |  |
| --- | --- |
| Syntax Sheet | Actual Code |
|  | UPPERCASE: SQL keywords  User named objects/values (ex. Table names, constraint names, column names…)  Data types  Strings/ data the user has decided to enter |
| CREATE TABLE table\_name  (  {<column\_definition> | <table\_constraint> } [ ,...n ]  )  <column\_definition> ::=  column\_name data\_type  [NULL | NOT NULL]  [[CONSTRAINT constraint\_name] DEFAULT constant\_expression]  | [IDENTITY [(seed, increment)]  ]  [<column\_constraint> [ ,...n ]]  <column\_constraint> ::=  CONSTRAINT constraint\_name  { {PRIMARY KEY | UNIQUE} [CLUSTERED | NONCLUSTERED]  | [FOREIGN KEY] REFERENCES referenced\_table\_name (ref\_column)  | CHECK (logical\_expression)  }  < table\_constraint > ::=  CONSTRAINT constraint\_name  {  {PRIMARY KEY | UNIQUE} [CLUSTERED | NONCLUSTERED] (column\_name [ ,...n ])  | FOREIGN KEY (column\_name [ ,...n ])  REFERENCES referenced\_table\_name (ref\_column [ ,...n ])  | CHECK (logical\_expression)  } | CREATE TABLE customer  (  CustomerID int NOT NULL  IDENTITY (100, 1)  CONSTRAINT pk\_Customer PRIMARY KEY CLUSTERED,  FirstName varchar(40) NOT NULL,  LastName varchar(50) NOT NULL,  Phone varchar(14) NOT NULL  CONSTRAINT ck\_Customer\_ValidPhone  CHECK (Phone LIKE '[1-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'),  Address varchar(30) NOT NULL,  City varchar(20) NOT NULL  CONSTRAINT df\_Customer\_City DEFAULT 'Edmonton',  PostalCode varchar(7) NOT NULL  CONSTRAINT ck\_Customer\_PostalCodeFormat  CHECK (PostalCode like '[A-Z][0-9][A-Z][0-9][A-Z][0-9]’),  Province char(2) NOT NULL  CONSTRAINT df\_Customer\_Province DEFAULT 'AB'  CONSTRAINT ck\_Customer\_ProvinceValid CHECK (Province like '[A-Z][A-Z]'),  Country varchar(15) NOT NULL  CONSTRAINT df\_Customer\_Country DEFAULT 'Canada'  CONSTRAINT ck\_tableLevelExample CHECK (Country != Province)  CONSTRAINT pk\_tableLevelPK PRIMARY KEY CLUSTERED (FirstName, LastName)  ) |
| ALTER TABLE table\_name  { [{check | nocheck} CONSTRAINT constraint\_name]  | [with {CHECK | NOCHECK}]  {{ADD {<column\_definition> | <table\_constraint>}  | ALTER COLUMN <column\_definition>}  | drop {COLUMN column\_name | CONSTRAINT constraint\_name}  } | ALTER TABLE Customer  CHECK CONSTRAINT ck\_Customer\_ValidPhone  ALTER TABLE Customer  ADD Email varchar(40) NULL  CONSTRAINT ck\_Customer\_EmailFormat CHECK (Email like '%\_\_@\_\_%.\_\_%')  ALTER TABLE Customer  ADD CONSTRAINT ck\_FirstNotLast CHECK (FirstName != LastName)  ALTER TABLE Customer  DROP COLUMN Phone  ALTER TABLE Customer  DROP CONSTRAINT ck\_FirstNotLast |
|  |  |

(this is meant to be tiny so you can put it on the board and compare them)