

DDL: Defining the database

Database Development 2021
Cody Henrichsen

Data Definition Language

- ▷ Making
- ▷ Changing
- ▷ Destroying
- ▷ All DDL subset commands depend on the users role or permission level

CREATE

- ▷ Used to make the components used in a database
 - ▷ DATABASE, TABLE, VIEW are the main structures
 - ▷ ROLE and USER are for permissions
 - ▷ INDEX, FUNCTION, PROCEDURE, TRIGGER are tools within the database

CREATE TABLE Syntax

- ▶ Wrapped in parentheses
- ▶ One line per field
- ▶ Name Type any modifiers
- ▶ Separate each line with a comma
- ▶ End with our great friend the semicolon

CREATE Example

```
CREATE TABLE table_name
(  
    field_name TYPE(parameter) NOT NULL,  
    other_field_name TYPE  
    --ONE FIELD PER LINE!!!!  
    --Indentation is important  
    --for humans to read  
);
```

ALTER

- ▶ This is how you change the structure of the table
- ▶ Should ONLY be done VERY early as it will cause data integrity violations
- ▶ Best practice
 - ▷ Adding PRIMARY and FOREIGN KEY to a table
- ▶ BAD practice
 - ▷ Changing column count

ALTER Primary

```
ALTER TABLE
  table_name
ADD PRIMARY KEY
  (key_field);

ALTER TABLE
  demo_table
ADD PRIMARY KEY
  (composite_key, other_key);
```

ALTER Foreign

```
ALTER TABLE
  demo_table
ADD CONSTRAINT
  fk_reference_source
FOREIGN KEY
  (key_field)
REFERENCES
  table_name (key_field)
ON DELETE
  CASCADE;
```

FOREIGN KEY example

```
ALTER TABLE
  sales
ADD CONSTRAINT
  fk_exporters_sales -- Name the relationship
FOREIGN KEY
  (export_id) -- ID the FIELD in parens
REFERENCES
  exporters (export_destination_code)
  -- table (primary_key) Note the names of the columns DO NOT HAVE TO MATCH!!!
ON DELETE
  CASCADE ;
```

DROP

- ▶ This is the command to destroy structures and the associated data
- ▶ No prompt or warning!
- ▶ Only way to get error is if you give an invalid data source!

DROP Syntax

```
• DROP TABLE  
  table_name;
```

```
• DROP TABLE IF EXISTS  
  table_name;
```
