**Tables List\_Heroku-Postgres db**

10 tables

6 rows:

**Shopping Cart tables**

1. TblOrders

CREATE TABLE orders (

order\_id serial PRIMARY KEY,

order\_date date NOT NULL DEFAULT CURRENT\_TIMESTAMP,

order\_name varchar(255) NOT NULL,

order\_email varchar(255) NOT NULL

);

1. tblProducts (6) rows

CREATE TABLE products (

product\_id serial PRIMARY KEY,

product\_name varchar(255) NOT NULL,

product\_image varchar(255) DEFAULT NULL,

product\_description text,

product\_price decimal(10,2) NOT NULL DEFAULT 0.00

);

INSERT INTO products (product\_name , product\_image , product\_description , product\_price ) VALUES

('Car' , 'car.jpg' , 'It''s a car. Batteries not included, not required. Powered by hand.' , 6.00 ),

('Dangerous Bear' , 'dangerous-bear.jpg' , 'Beware. This bear is extremely dangerous.' , 8.00 ),

('Fish' , 'fish.jpg' , 'There is something fishy going on here... ', 7.50 ),

('Chill Gorilla' , 'gorilla.jpg' , 'Unlike the dangerous bear, this one is chill.' , 8.80 ),

('Rubber Duck' , 'rubber-duck.jpg' , 'Best partner in the bath tub.' , 9.75 ),

('Rubiks Cube' , 'rubiks-cube.jpg' , 'Some say that this cube trains your intelligence. Some others claim that it''s just frustration.' , 9.30 );

1. tblOrders\_items

CREATE TABLE orders\_items (

order\_id serial PRIMARY KEY REFERENCES orders(order\_id),

product\_id int NOT NULL REFERENCES products(product\_id),

quantity int NOT NULL

);

**Scripture tables**

1. tblScriptures (4) rows

CREATE TABLE scriptures (

scriptures\_id serial PRIMARY KEY,

scriptures\_book VARCHAR(40),

scriptures\_chapter INT,

scriptures\_verse INT,

scriptures\_content varchar(1000)

);

--Add Values to Scriptures Table (W05 Teach

INSERT INTO scriptures(scriptures\_book, scriptures\_chapter, scriptures\_verse, scriptures\_content)

VALUES ('John', 1, 5, 'And the light shineth in darkness; and the darkness comprehended it not.'),

('Doctrine and Covenants', 88, 49, 'The light shineth in darkness, and the darkness comprehendeth it not; nevertheless, the day shall come when you shall comprehend even God, being quickened in him and by him.'),

('Doctrine and Covenants', 93, 28, 'He that keepeth his commandments receiveth truth and light, until he is glorified in truth and knoweth all things.'),

('Mosiah', 16, 9, 'He is the light and the life of the world; yea, a light that is endless, that can never be darkened; yea, and also a life which is endless, that there can be no more death.');

**Conference Talk tables**

1. tblUser\_table

CREATE TABLE user\_table (

user\_id serial PRIMARY KEY,

username varchar(255) UNIQUE NOT NULL,

user\_password varchar(255) NOT NULL,

user\_email varchar(255) UNIQUE NOT NULL,

created\_on TIMESTAMP NOT NULL,

last\_login TIMESTAMP

);

1. tblSpeakers

CREATE TABLE speakers (

speaker\_id serial PRIMARY KEY,

speaker\_name varchar(255) NOT NULL

);

1. tblTalk

CREATE TABLE talk (

talk\_id serial PRIMARY KEY,

speaker\_id int NOT NULL REFERENCES speakers(speaker\_id),

talk\_date date NOT NULL,

talk\_session varchar(255) NOT NULL,

talk\_url varchar(255) NOT NULL

);

1. tblNotes

CREATE TABLE notes (

note\_id serial PRIMARY KEY,

user\_id int NOT NULL REFERENCES user\_table(user\_id),

talk\_id int NOT NULL REFERENCES talk(talk\_id),

notes\_text text NOT NULL

);

1. tblNote\_user (2 rows)

CREATE TABLE note\_user (

id SERIAL,

username VARCHAR(255),

password VARCHAR(255),

PRIMARY KEY (id)

);

INSERT INTO note\_user (username, password)

VALUES ('john', 'pass');

INSERT INTO note\_user (username, password)

VALUES ('jane', 'byui');

1. tblNote (3 rows)

CREATE TABLE note (

id SERIAL,

userId INT NOT NULL,

content TEXT,

PRIMARY KEY (id),

FOREIGN KEY (userId) REFERENCES note\_user (id)

);

INSERT INTO note (userId, content) VALUES (1, 'A note for John');

INSERT INTO note (userId, content) VALUES (1, 'Another note for John');

INSERT INTO note (userId, content) VALUES (2, 'And this is a note for Jane');

New postgres db flexible-81262 – created on 2/7/19 with tables and data above

A screenshot of a computer

Description automatically generated

