<u>Team - Group 18</u>:
Angel Tapia (ONID: tapiaang)
Ryan Alexandra Davis (ONID: davisrya)

Project Step 2: ERD & Schema (Draft)

Part A: Fixes based on Feedback from Step 1

Our feedback was mostly about our overview, saying that this could be a bit more focused on the reason for needing a database-driven website. Also, since this is going to be used mainly for the business and not really customer-facing it would have been helpful to know how more numerical facts related to the entities. We have rewritten our overview to more closely align with the grader's advice.

There was also a note on the amount of M:M relationships we currently have, and how that would be difficult to implement. We have decided on what to eliminate.

Part B: Project Outline and Database Outline - Updated Version

Notes:

Our *order* and *keyboard* entities will be the only M:M relationship in our final design. We originally had multiple M:M relationships by assuming that a keyboard could have multiple kinds of switches and key color combinations. This will not be the case for our final design. *switchComp* and *colorComp* will be removed in the final design, resulting in our team implementing 6 entities instead of 8.

Division of Work:

- Angel will primarily be in charge of the M:M relationship, which will involve working with the *order*, *keyboard*, and *keyboardOrderComp* entities.
- Ryan will primarily be in charge of the *customer*, *switch*, and *keyColor* entities.

Project: "Definitely *Not* a Cult"

Mechanical keyboards are some of the most customizable computer peripherals available today. Many enthusiasts enjoy building their own mechanical keyboards, but newcomers to the hobby may be intimidated by the infinite possibilities of customizable parts including, but not limited to, mechanical switches and keycaps.

“Definitely *Not* a Cult” is a growing online business specializing in pre-built yet modifiable beginner keyboards to help initiate newcomers into the world of mechanical keyboards. They currently offer six different mechanical switch and keycap color combinations per keyboard in each order but plan to increase the selection of customizable parts in the near future.

Currently they are selling approximately 7,500 keyboards annually with 4,000 unique customers having bought their products, and now they are aiming to raise these numbers. However, their current pain point is keeping their customers’ orders clearly and concisely organized: a business run by three individuals leaves room for disorganization between physically building orders and keeping track of the business aspects manually with GoogleDrive. An actual database focusing on retaining and updating unique customer and order information would allow “Definitely *Not* a Cult” to thrive and do what they do best: building keyboards and spreading the joy of the mechanical keyboard cult.

customer

Table holding basic information about the customer
Variables:

- *customerNum*: INT, auto_increment, unique, not NULL, PRIMARY_KEY
- *lastName*: VARCHAR, not NULL
- *firstName*: VARCHAR, not NULL
- Relationship: a 1:M relationship between Customers and Orders is implemented with customerNum as a FK inside of Orders

order

Table containing unique order information

- *orderNum*: INT, auto_increment, unique, not NULL, PRIMARY_KEY
- *customerNum*: INT, not NULL, FOREIGN_KEY
- *orderDate*: DATE, not NULL
- *paypalAccount*: VARCHAR, not NULL
    - This is expected to be an email
- Relationship: a 1:M relationship between Orders and keyboardOrderComp is implemented with orderNum as a FK inside of keyboardOrderComp

keyboardOrderComp

Composite table connecting Keyboard and Order (organizes M:M relationships)

- *orderNum*: INT, not NULL, FOREIGN_KEY
- *keyboardNum:* INT, not NULL, FOREIGN_KEY
- *quantityOrdered*: INT, not NULL
- *pricePerUnit*: DECIMAL(6,2), not NULL
- Relationship: a 1:M relationship between keyboardOrderComp and Keyboard is implemented with keyboardNum as a FK inside of keyboardOrderComp

keyboard

Table containing details on each pre-built keyboard variation

Variables:

- *keyboardNum*: INT, auto_increment, unique, not NULL, PRIMARY_KEY
  - This refers to the type of fully built keyboard with chosen components, not an individual keyboard.
- *name*: VARCHAR, not NULL
  - Shortened version of the components in the keyboard
- *quantityInStock*: INT, not NULL
- Relationship: a 1:M relationship between Keyboard and switchComp is implemented with keyboardNum as a FK inside of switchComp

switchComp

Composite table connecting Keyboard and Switches (organizes M:M relationships)

Variables:

- *switchNum*: INT, not NULL, FOREIGN_KEY
- *keyboardNum:* INT, not NULL, FOREIGN_KEY
- Relationships: a 1:M relationship between switchComp and Switches is implemented with switchNum as a FK inside of switchComp. A 1:M relationship between colorComp and Keyboard is implemented with keyboardNum as a FK inside of colorComp

switch

Table that holds information regarding switch details

Variables:

- *switchNum*: INT, not NULL, PRIMARY_KEY
- *switchName:* VARCHAR, not NULL
- Relationship: Same as in SwitchComp

colorComp

Composite table connecting Keyboard and KeyColors (organizes M:M relationships)
Variables:

- *keyColorNum*: INT, not NULL, FOREIGN_KEY
- *keyboardNum:* INT, not NULL, FOREIGN_KEY
- Relationship: a 1:M relationship between colorComp and KeyColors is implemented with keyColorNum as a FK inside of colorComp

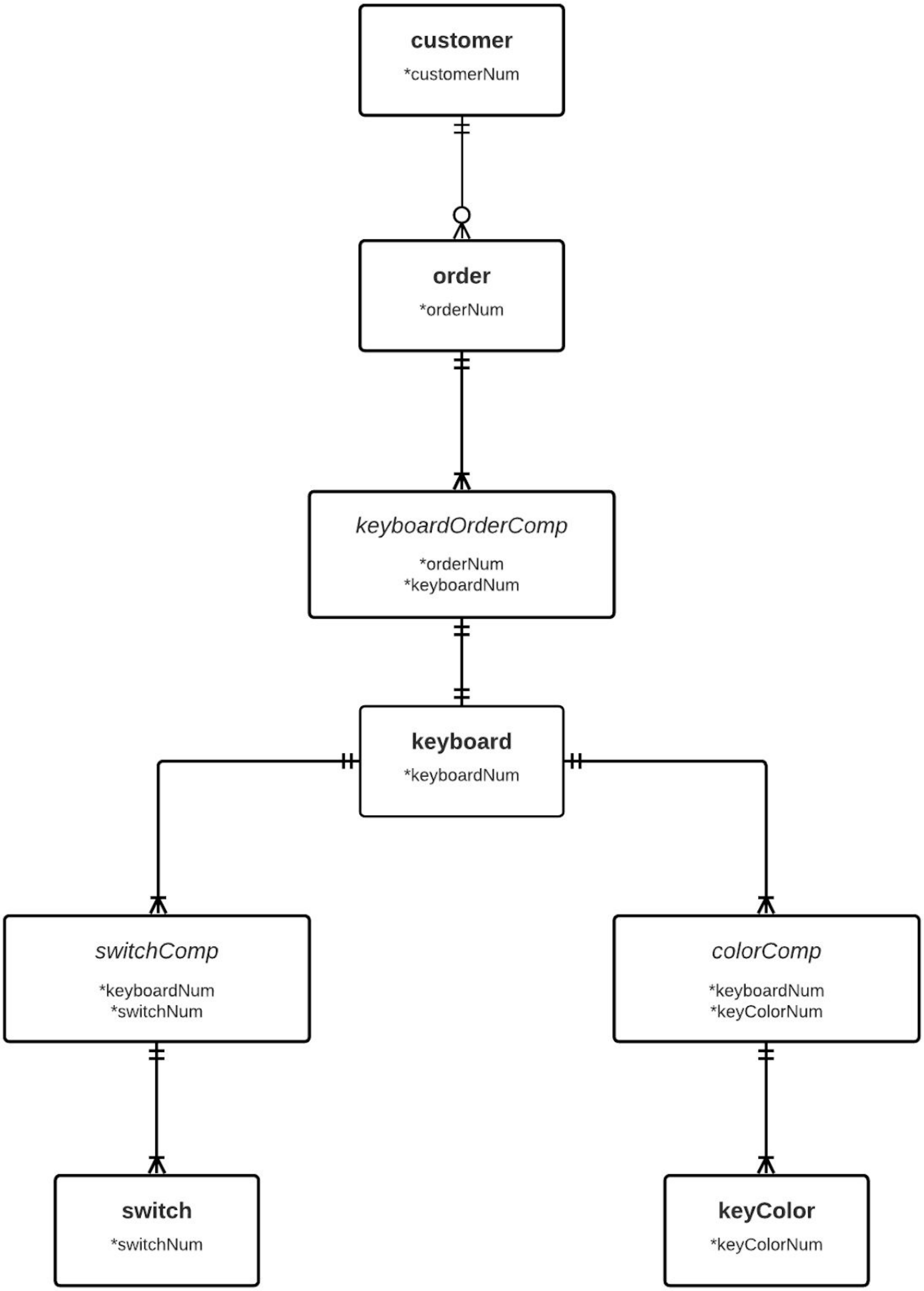
keyColor

Table that holds information regarding colors of keycaps
Variables:

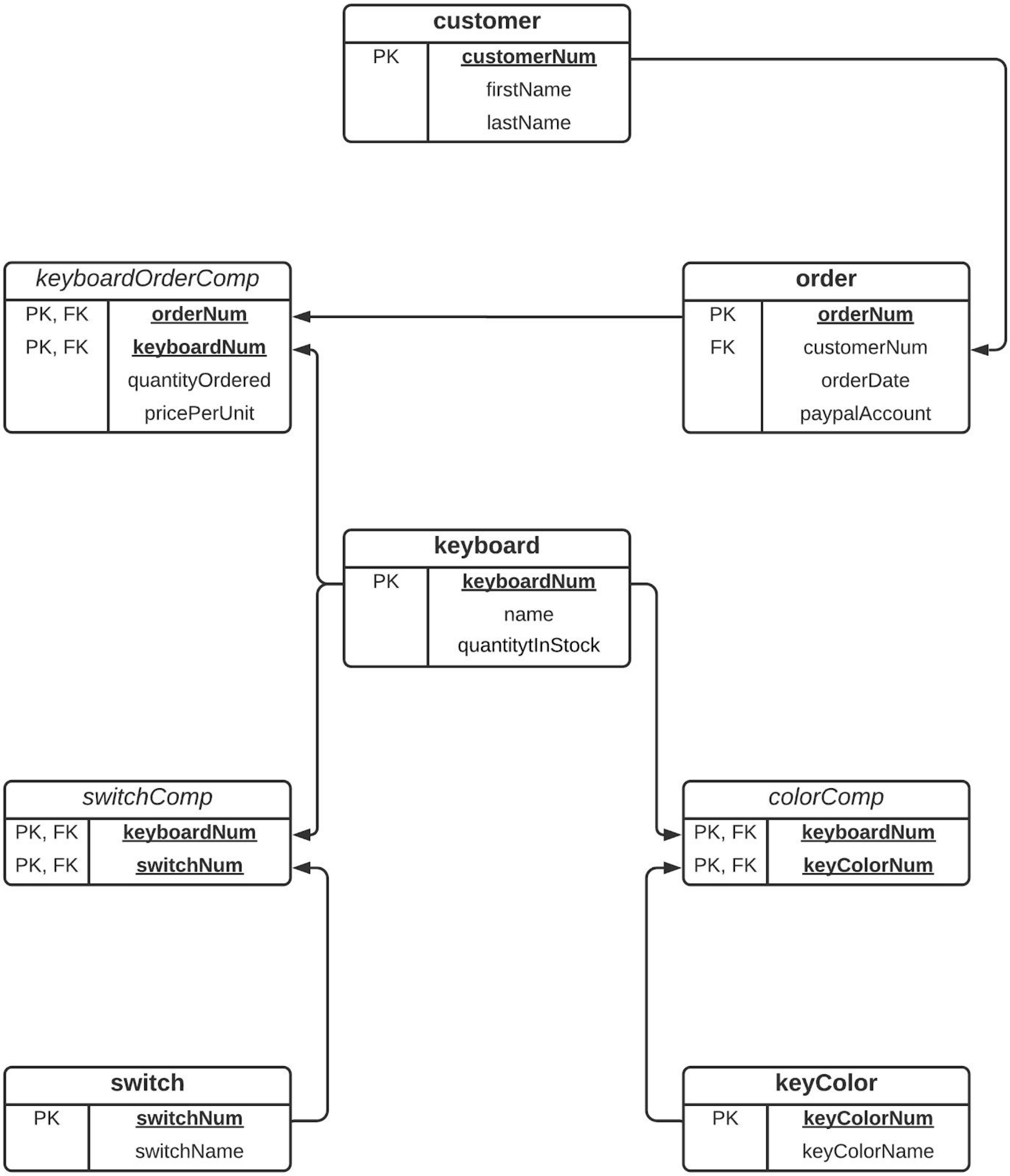
- *keyColorNum*: INT, not NULL, PRIMARY_KEY
- *keyColorName:* VARCHAR, not NULL
- Relationship: Same as in colorComp

Parts C and D can be found starting on the next page.

Part C: Entity-Relationship Diagram



Part D: Schema


customer
PK
customerNum
firstName
lastName
keyboardOrderComp
PK, FK
orderNum
PK, FK
keyboardNum
quantityOrdered
pricePerUnit
order
PK
orderNum
FK
customerNum
orderDate
paypalAccount
keyboard
PK
keyboardNum
name
quantitytInStock
switchComp
PK, FK
keyboardNum
PK, FK
switchNum
colorComp
PK, FK
keyboardNum
PK, FK
keyColorNum
switch
PK
switchNum
switchName
keyColor
PK
keyColorNum
keyColorName