Database Design

The Physical Model

OVERVIEW OF THE FIVE DATABASE SESSIONS

- Session 1: The Transactional Relational Database
    - Work product: Conceptual Model
- Session 2: Normalizing the Transactional Relational Database
    - Work product: Logical Model
- **Session 3: Defining Data Structures Specific To A Database Platform (MariaDB)**
    - **Work product: Physical Model**
- Session 4: Database Initialization Scripts To Create Databases & Objects
    - Work product: SQL Scripts To Create Database Objects
- Session 5: SQL Essentials To Query Databases
    - Work product: SQL Commands To Query The Database

First, Let’s Review

- What is the purpose of the conceptual model?
  - It must be understood by everyone
- What is the purpose of the logical model?
  - To show the primary key and non-key attributes of each entity
  - To show the relationship between the entities using foreign keys.
- How is logical model similar to the conceptual model?
  - They both are platform agnostic.
- How is logical model different from the conceptual model?
  - It does not need to be understandable by everyone.

Session 3 Objectives

- Why bother with a physical model?
- What is a key difference of the physical model from both the conceptual and logical model?
- Choose Database Engine.
- Common Data Types and when to use them.
- What is a default value?
- What is Null?
- What is a check constraint?

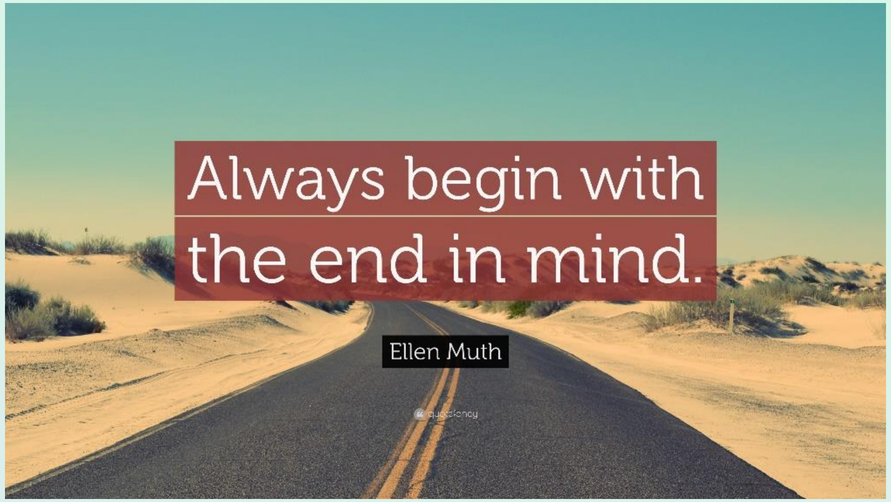
Why Bother With A Physical Model

- A physical data model describes how data is organized in actual database tables.
- They provide additional attributes not specified in a logical data model, such as data types.
- They map the data elements to an actual database data type.

Physical Model — A Key Difference

- The Conceptual and Logical Models are both platform agnostic.
  - These are the same regardless of the final database platform.

- The Physical Model is aimed at a particular database platform.
  - Therefore the final product will be different from one platform to another.

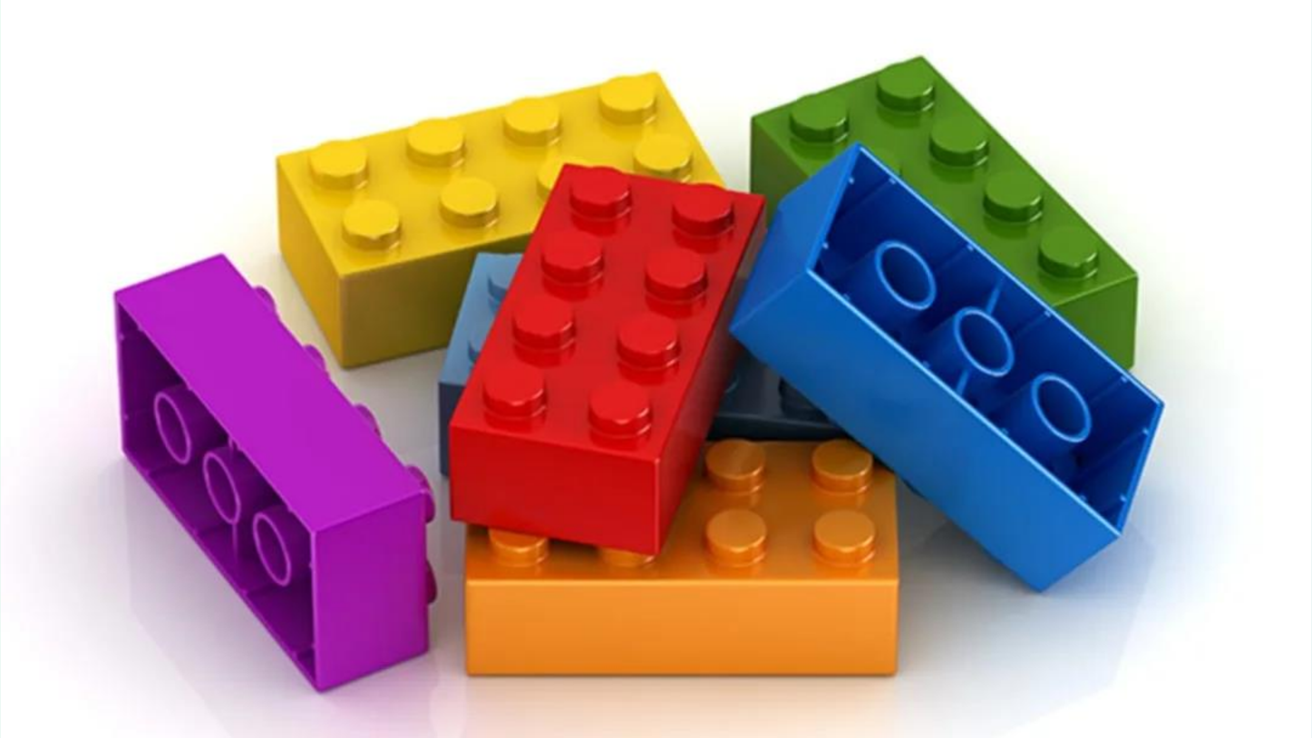


Physical Model — Choose Database Engine

- The cook book — the Logical model to a Physical model.
- Use database terms instead of logical model terms
  - Tables instead of entities
  - Columns / fields instead of attributes
- Identify data types and sizes

Basic Data Types

- Text
- Numbers
- Date / Time
- Boolean

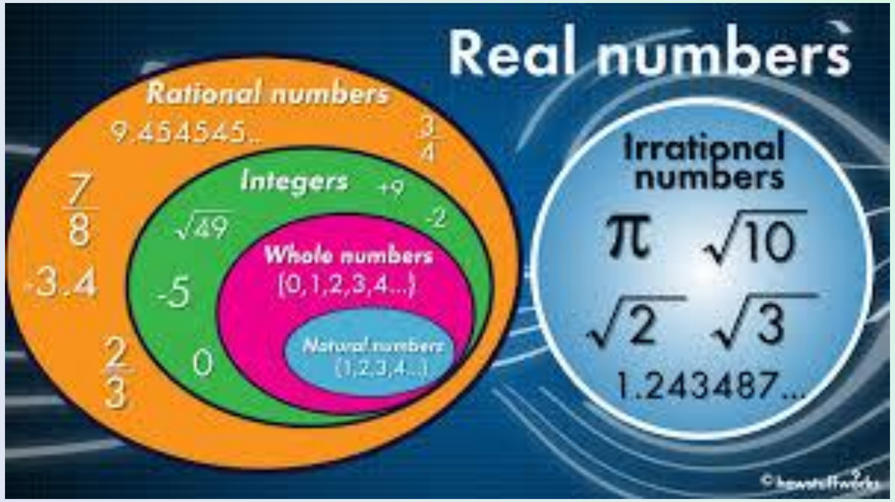


Basic Data Types — **Text**

- String Values
    1. Character{**x**}: fixed lengths
    2. Varchar {**x**}: variable lengths
    3. Nchar{**x**}: double byte
- "**x**" (if used) should be the smallest value that can hold the largest value that you can reasonably expect to be inserted.
    - Too many folks use big numbers without a thought. (= bad data governance)
- Sometimes, just because you can, does not mean you should.

Basic Data Types — **Numbers**

- Integer – (Tinyint, Smallint, Integer, Bigint)
- Numeric – (Decimal, Float, Double) {M,D}
    - “M” (if used) is the total number of digits (the precision)
    - “D” (if used) is the number of digits after the decimal point (the scale)



Basic Data Types — **Dates and Times**

- Date (YYYY-MM-DD)
- DateTime (YYYY-MM-DD HH:MM:SS.ssssss)
- Time (HH:MM:SS.ssssss)
- TimeStamp (YYYY-MM-DD HH:MM:SS.ssssss)

Basic Data Types — **Misc**

- Boolean
- Timezone

Basic Data Types — **MariaDB specific**

- Understand Database Engine limitations
- **Timezone**
  - Any inserted values are converted from the session's time zone to **UTC** (the world standard for regulating time) when stored, and converted back to the session's time zone when retrieved
- **Timestamp**
  - Can hold values between '1970-01-01 00:00:01' (**UTC**) and '2038-01-19 03:14:07' (**UTC**)
- **DateTime**
  - Due to Timestamp restrictions, DateTime should be used instead (*same format but no data value restrictions*)
- **Boolean**
  - Defined as TINYINT(1).
  - A value of zero is considered false. All non-zero values are considered true.

Default Values

- Columns can be assigned a default value

Name	Address1	Address2	City	State	Zip
Winsupply	3110 Kettering Blvd	Null	Dayton	OH	45439
Winsupply	3110 Kettering Blvd	“”	Dayton	OH	45439

Null Values

- Null means the value is unknown or unknowable
- A Null value is not the same as any other value
  - not even another Null value.

Name	Address1	Address2	City	State	Zip
Winsupply	3110 Kettering Blvd	Null	Dayton	OH	45439
Winsupply	3110 Kettering Blvd		Dayton	OH	45439

Examples Of When To Use Null Or A Blank

- Null means the value is unknown
- A Blank can mean the value does not exist
  - *Blanks only apply to text data types; numbers and dates cannot have blanks*
- Sometimes null is correct and sometimes it is not.
- Sometimes a blank instead of a null can “break” applications/code though.

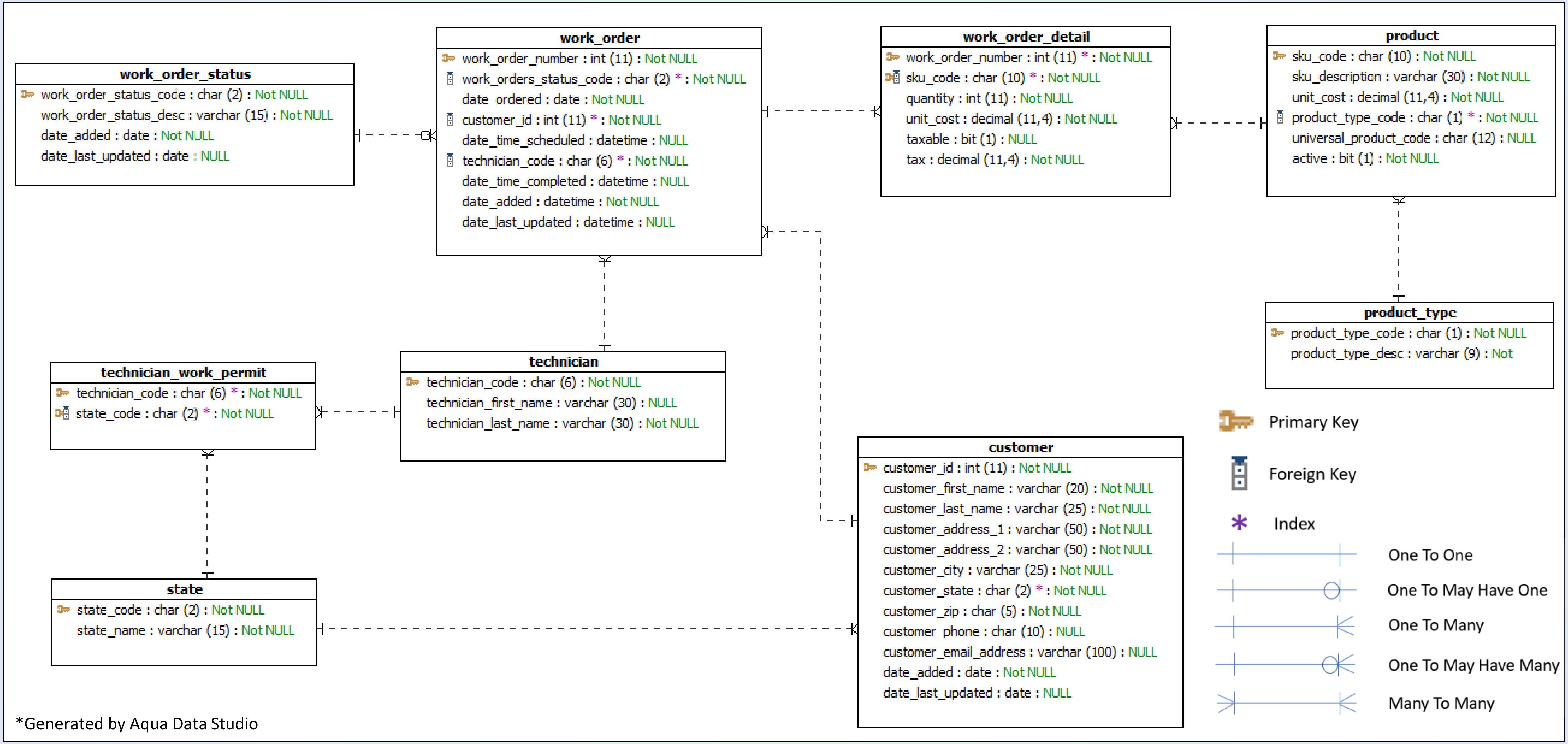
- When To Use Null:
  - 1. newborn baby SS # – this is unknown until it is applied for
  - 2. zip + extension – addresses have zip + 4 but the extension may be unknown
- When To Use Blank:
  - 1. address 2 – blank when this is known to be true
  - 2. middle name – when a person does not have a middle name and it is required

Check Constraints

- Column level
  - A value must fall with in a certain range… [A,B,C] or [T,F] or [Active,Inactive]
- Table level
  - A column’s value has some dependence on another column’s value. A > B

- It’s all about data quality.

WORK ORDER PRO – PHYSICAL DATA MODEL



work_order_status
work_order_status_code : char (2) : Not NULL
work_order_status_desc : varchar (15) : Not NULL
date_added : date : Not NULL
date_last_updated : date : NULL

work_order
work_order_number : int (11) : Not NULL
work_orders_status_code : char (2) * : Not NULL
date_ordered : date : Not NULL
customer_id : int (11) * : Not NULL
date_time_scheduled : datetime : NULL
technician_code : char (6) * : Not NULL
date_time_completed : datetime : NULL
date_added : datetime : Not NULL
date_last_updated : datetime : NULL

work_order_detail
work_order_number : int (11) * : Not NULL
sku_code : char (10) * : Not NULL
quantity : int (11) : Not NULL
unit_cost : decimal (11,4) : Not NULL
taxable : bit (1) : NULL
tax : decimal (11,4) : Not NULL

product
sku_code : char (10) : Not NULL
sku_description : varchar (30) : Not NULL
unit_cost : decimal (11,4) : Not NULL
product_type_code : char (1) * : Not NULL
universal_product_code : char (12) : NULL
active : bit (1) : Not NULL

product_type
product_type_code : char (1) : Not NULL
product_type_desc : varchar (9) : Not

technician_work_permit
technician_code : char (6) * : Not NULL
state_code : char (2) * : Not NULL

technician
technician_code : char (6) : Not NULL
technician_first_name : varchar (30) : NULL
technician_last_name : varchar (30) : Not NULL

state
state_code : char (2) : Not NULL
state_name : varchar (15) : Not NULL

customer
customer_id : int (11) : Not NULL
customer_first_name : varchar (20) : Not NULL
customer_last_name : varchar (25) : Not NULL
customer_address_1 : varchar (50) : Not NULL
customer_address_2 : varchar (50) : Not NULL
customer_city : varchar (25) : Not NULL
customer_state : char (2) * : Not NULL
customer_zip : char (5) : Not NULL
customer_phone : char (10) : NULL
customer_email_address : varchar (100) : NULL
date_added : date : Not NULL
date_last_updated : date : NULL

Primary Key

Foreign Key

Index

One To One

One To May Have One

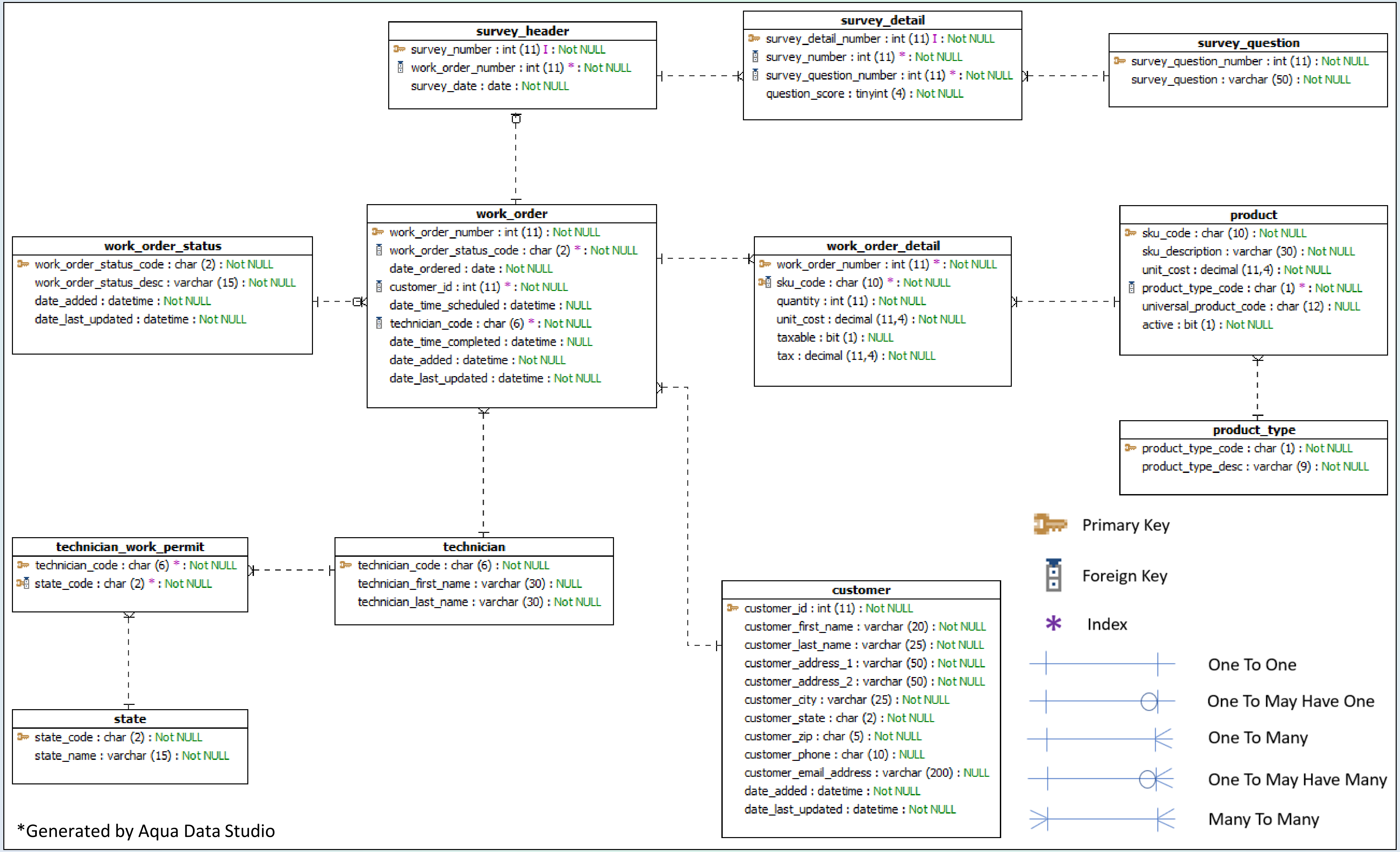
One To Many

One To May Have Many

Many To Many

*Generated by Aqua Data Studio

WORK ORDER PRO – PHYSICAL DATA MODEL – WITH SURVEY COMPONENT



survey_header
- survey_number : int (11) I : Not NULL
- work_order_number : int (11) * : Not NULL
- survey_date : date : Not NULL

survey_detail
- survey_detail_number : int (11) I : Not NULL
- survey_number : int (11) * : Not NULL
- survey_question_number : int (11) * : Not NULL
- question_score : tinyint (4) : Not NULL

survey_question
- survey_question_number : int (11) : Not NULL
- survey_question : varchar (50) : Not NULL

work_order_status
- work_order_status_code : char (2) : Not NULL
- work_order_status_desc : varchar (15) : Not NULL
- date_added : datetime : Not NULL
- date_last_updated : datetime : Not NULL

work_order
- work_order_number : int (11) : Not NULL
- work_order_status_code : char (2) * : Not NULL
- date_ordered : date : Not NULL
- customer_id : int (11) * : Not NULL
- date_time_scheduled : datetime : NULL
- technician_code : char (6) * : Not NULL
- date_time_completed : datetime : NULL
- date_added : datetime : Not NULL
- date_last_updated : datetime : Not NULL

work_order_detail
- work_order_number : int (11) * : Not NULL
- sku_code : char (10) * : Not NULL
- quantity : int (11) : Not NULL
- unit_cost : decimal (11,4) : Not NULL
- taxable : bit (1) : NULL
- tax : decimal (11,4) : Not NULL

product
- sku_code : char (10) : Not NULL
- sku_description : varchar (30) : Not NULL
- unit_cost : decimal (11,4) : Not NULL
- product_type_code : char (1) * : Not NULL
- universal_product_code : char (12) : NULL
- active : bit (1) : Not NULL

product_type
- product_type_code : char (1) : Not NULL
- product_type_desc : varchar (9) : Not NULL

technician_work_permit
- technician_code : char (6) * : Not NULL
- state_code : char (2) * : Not NULL

technician
- technician_code : char (6) : Not NULL
- technician_first_name : varchar (30) : NULL
- technician_last_name : varchar (30) : Not NULL

state
- state_code : char (2) : Not NULL
- state_name : varchar (15) : Not NULL

customer
- customer_id : int (11) : Not NULL
- customer_first_name : varchar (20) : Not NULL
- customer_last_name : varchar (25) : Not NULL
- customer_address_1 : varchar (50) : Not NULL
- customer_address_2 : varchar (50) : Not NULL
- customer_city : varchar (25) : Not NULL
- customer_state : char (2) : Not NULL
- customer_zip : char (5) : Not NULL
- customer_phone : char (10) : NULL
- customer_email_address : varchar (200) : NULL
- date_added : datetime : Not NULL
- date_last_updated : datetime : Not NULL

Primary Key

Foreign Key

* Index

One To One

One To May Have One

One To Many

One To May Have Many

Many To Many

*Generated by Aqua Data Studio

HOMEWORK – WRITE DATABASE CREATE SCRIPTS

- Session 1: The Transactional Relational Database
  - Work product: Conceptual Model
- Session 2: Normalizing the Transactional Relational Database
  - Work product: Logical Model
- Session 3: Defining Data Structures Specific To A Database Platform (MariaDB)
  - Work product: Physical Model
- **Session 4: Database Initialization Scripts To Create Databases & Objects**
  - **Work product: SQL Scripts To Create Database Objects**
- Session 5: SQL Essentials To Query Databases
  - Work product: SQL Commands To Query The Database