Databases Crash Course

Ignacio X. Domínguez

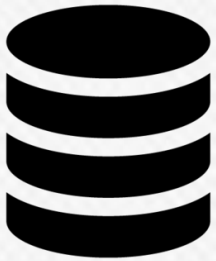
DiamondHacks 2021


NC STATE CSC

To Follow Along

- Make sure you have these running locally:
  - MySQL Community Server
  - MySQL Workbench

What is a Database?



- Structured data
  e.g. File system

- We use standard software to help us manage our data using common APIs

	A	B	C	D	E
1	College Enrollment 2016 - 2017				
2	Student ID	Last Name	Initial	Age	Program
3	ST348-245	White	R.	21	Drafting
4	ST348-246	Wilson	P.	19	Science
5	ST348-247	Thompson	A.	18	Arts
6	ST348-248	Holt	R.	23	Science
7	ST348-249	Armstrong	J.	37	Drafting
8	ST348-250	Graham	S.	20	Arts
9	ST348-251	McFadden	H.	26	Business
10	ST348-252	Jones	S.	22	Nursing
11	ST348-253	Russell	W.	20	Nursing
12	ST348-254	Smith	L.	19	Business

Database Engines (DBMS)

Database Management Systems

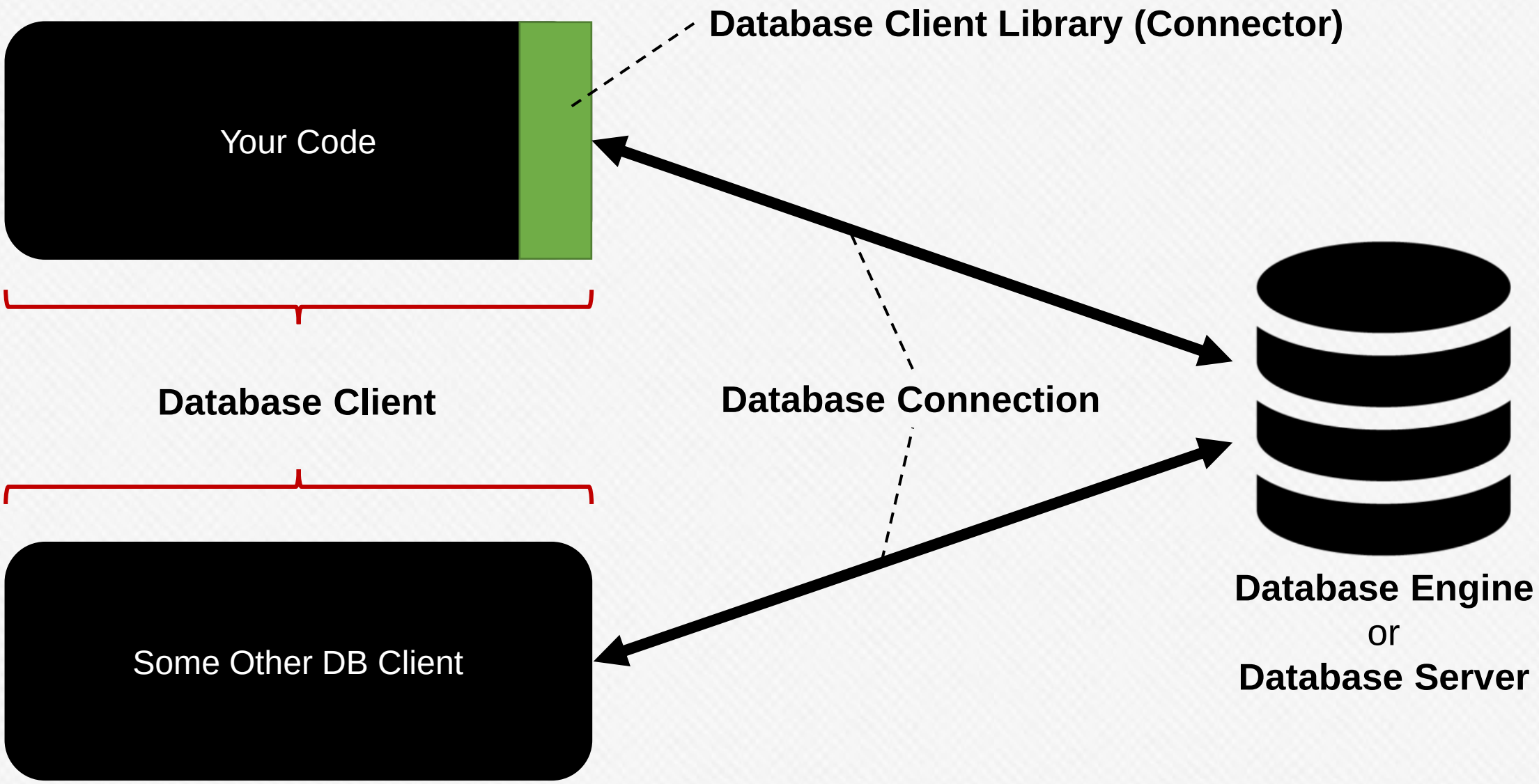
Relational engines (RDBMS)

- MySQL / MariaDB
- Microsoft SQL Server
- Oracle
- PostgreSQL
- SQLite
- …
- Others

Non-relational engines (NoSQL)

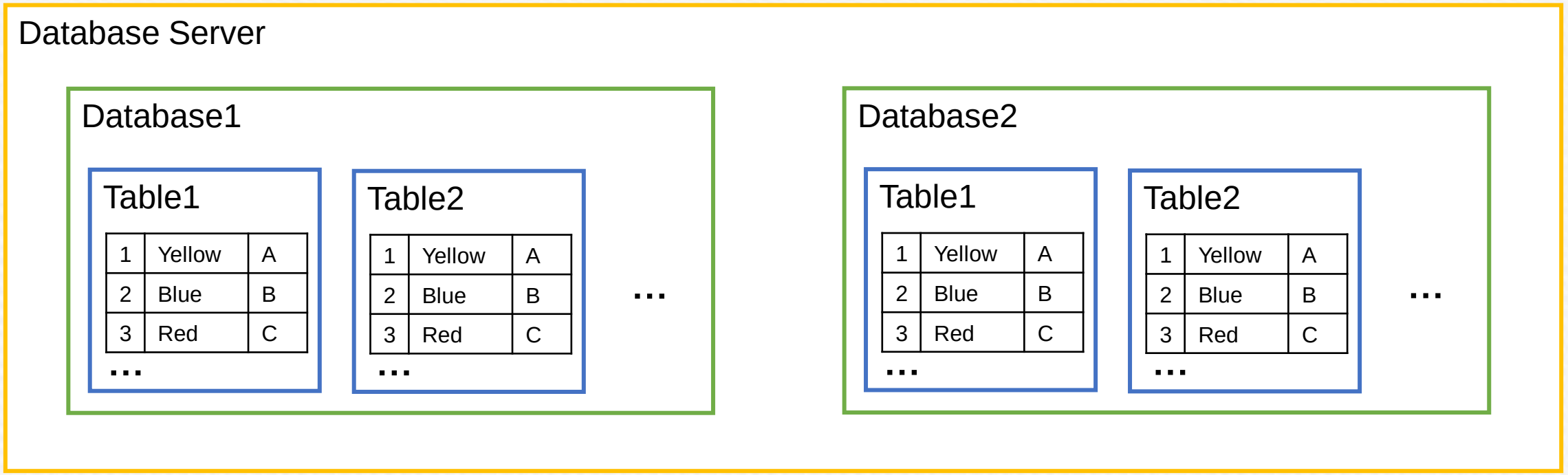
- Key-value stores (e.g., Redis, Berkeley DB, MemcacheD)
- Document (e.g., MongoDB, CouchDB, DocumentDB)
- Graph (e.g., Neo4j, Titan)
- …
- Others

Some Terminology



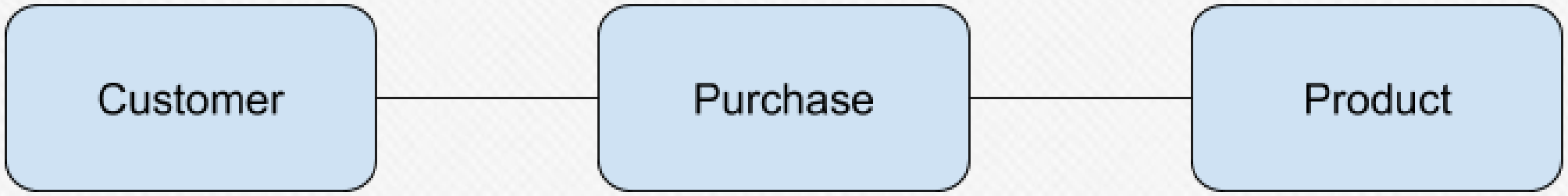
Some More Terminology

- A single ***database server*** can hold multiple ***databases*** (or ***schemas***)
- Each ***database*** can have multiple ***tables***
- Each ***table*** can have multiple ***records***



Example of Relational Data

Domain: A Store



SQL: Structured Query Language

- SQL is the language your code uses to manipulate a database

- CRUD:
  - Create `INSERT INTO customer (…) VALUES (…);`
  - Retrieve `SELECT * FROM customer;`
  - Update `UPDATE customer SET … WHERE …;`
  - Delete `DELETE FROM customer WHERE …;`

Tables (Entities)

- Table name
- Attributes (columns)
  - Names
  - Types
  - Length (precision)
  - Default value
  - Other constraints
- Stores multiple records (rows)

```
CREATE TABLE customer (
  cus_id          INT           NOT NULL AUTO_INCREMENT,
  cus_first_name  VARCHAR(45)   NOT NULL,
  cus_last_name   VARCHAR(45)   NULL DEFAULT NULL,
  cus_email       VARCHAR(200)  NOT NULL,
  PRIMARY KEY (cus_id)
);
```

cus_id	cus_first_name	cus_last_name	cus_email
1	Jane	Doe	jane@email.com
2	John	*(null)*	john@email.com
…			

Primary Keys

- Uniquely identifies a record
  Not null + unique
- Natural key
  e.g., room number in a building
- Artificial key
  e.g., auto-incremented number

number	max_occupancy
100	4
101	4
200	2
201	2

Relations: Foreign Keys

Referential Integrity

- External private key as column
- Can be `NULL`

customer

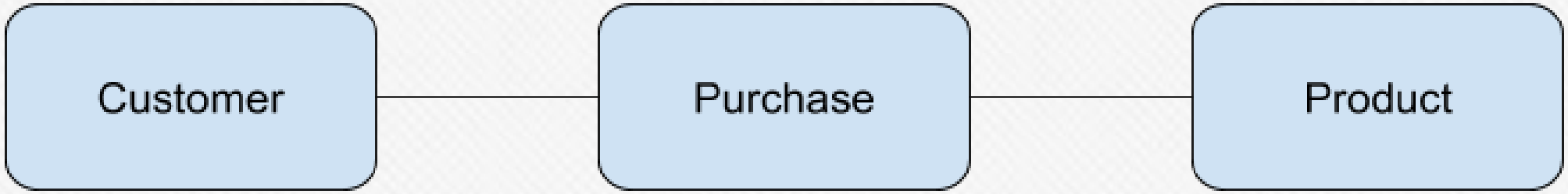
id	first	last
1	Alice	Doe
2	Bob	Doe
3	Jane	Doe
4	John	Doe

purchase

id	cust_id	date	total
1	1	2021-03-27	35.99
2	1	2021-03-27	1.23
3	3	2021-03-27	1.00
4	2	2021-03-27	3.99
5	2	2021-03-27	2.75
6	1	2021-03-27	0.49

```
ALTER TABLE purchase ADD CONSTRAINT fk_purchase_customer
    FOREIGN KEY (cust_id) REFERENCES customer (id);
```

Creating Our Schema



Id
First Name
Last Name
Email

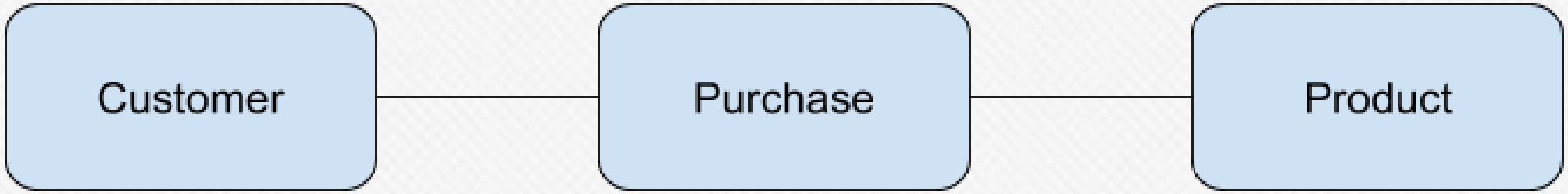
Id
Customer Id
Date
Total

Products?

Id
Name
Description
Price

id	cust_id	date	total
1	1	2021-03-27	35.99
2	1	2021-03-27	1.23
3	3	2021-03-27	1.00
4	2	2021-03-27	3.99
5	2	2021-03-27	2.75
6	1	2021-03-27	0.49

Creating Our Schema



Id
First Name
Last Name
Email

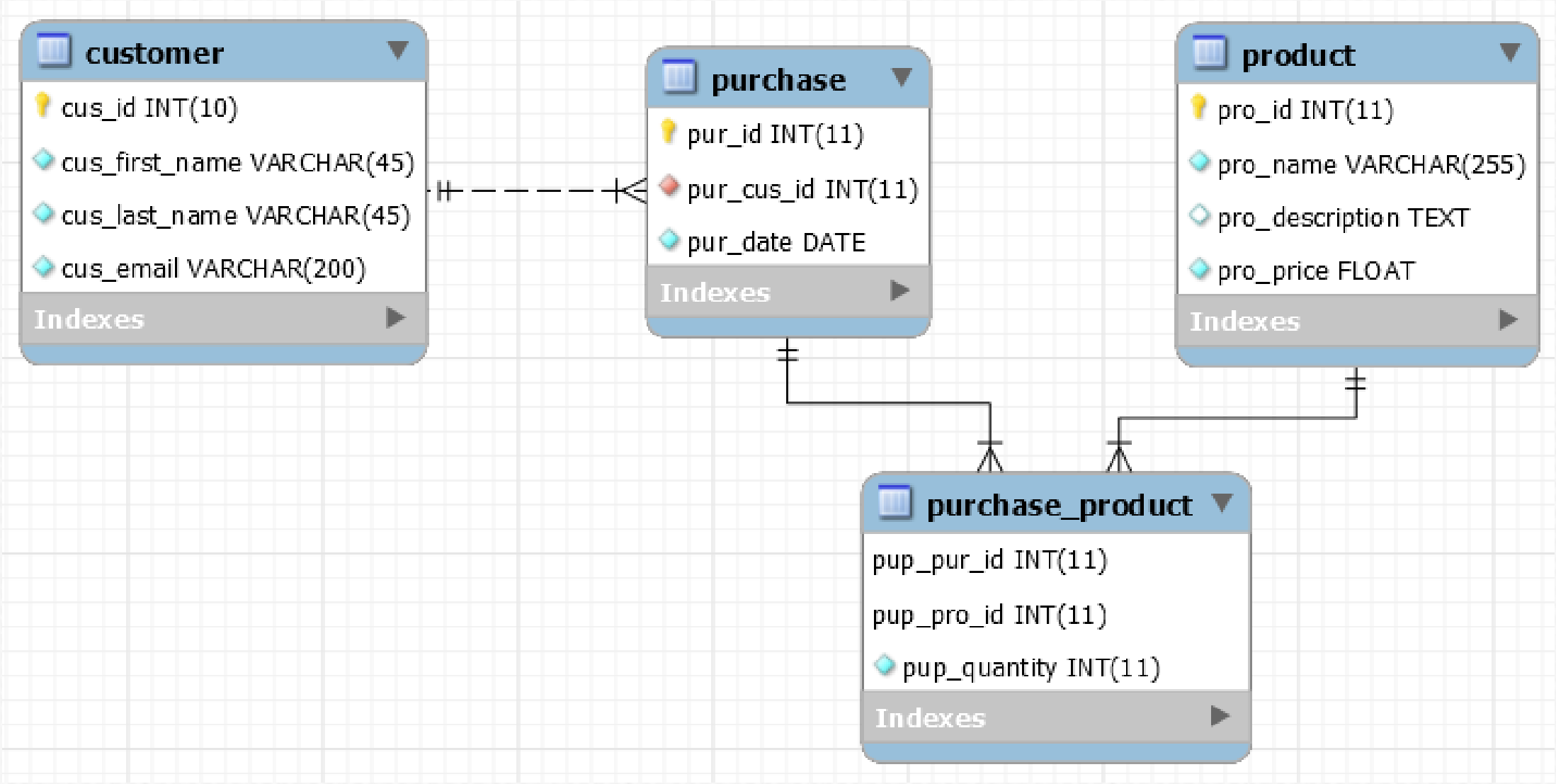
Id
Customer Id
Date
~~Total~~

Id
Name
Description
Price

id	cust_id	date	~~total~~
1	1	2021-03-27	~~35.99~~
2	1	2021-03-27	~~1.23~~
3	3	2021-03-27	~~1.00~~
4	2	2021-03-27	~~3.99~~
5	2	2021-03-27	~~2.75~~
6	1	2021-03-27	~~0.49~~

Purchase Product

Purchase Id
Product Id
Quantity


customer
cus_id INT(10)
cus_first_name VARCHAR(45)
cus_last_name VARCHAR(45)
cus_email VARCHAR(200)
Indexes
purchase
pur_id INT(11)
pur_cus_id INT(11)
pur_date DATE
Indexes
product
pro_id INT(11)
pro_name VARCHAR(255)
pro_description TEXT
pro_price FLOAT
Indexes
purchase_product
pup_pur_id INT(11)
pup_pro_id INT(11)
pup_quantity INT(11)
Indexes

Using Your DB in Your Code

Case Study: Python

1. Import MySQL Connector for Python
   - https://github.com/mysql/mysql-connector-python
2. Establish the connection to the DB
   - Host, port, and schema
   - Username and password
3. Get a “cursor”
4. Execute a query
5. Read response from the cursor
   - In this case, retrieve the first row
   - Each row is an array of all columns
6. Close the connection

```
import mysql.connector

# Connect to server
cnx = mysql.connector.connect(
    host="127.0.0.1", port=3306, database='db_workshop',
    user="root", password="super_s3cret")

# Get a cursor
cur = cnx.cursor()

# Execute a query
cur.execute("SELECT * FROM customer")

# Fetch one result
row = cur.fetchone()
print("Customer's first name is: {0}".format(row[1]))

# Close connection
cnx.close()
```

Let’s Hop on Over to MySQL Workbench

Feel free to follow along