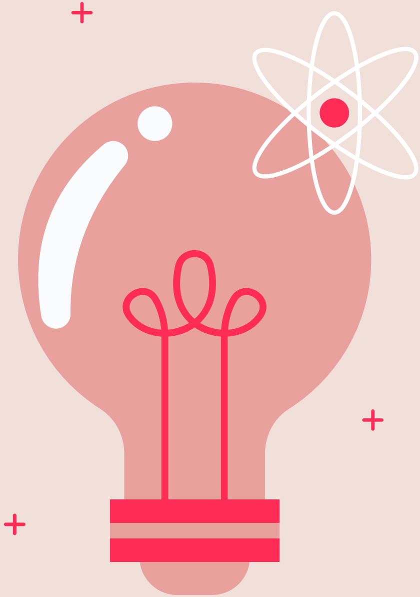
# Intro to SQL



# 01

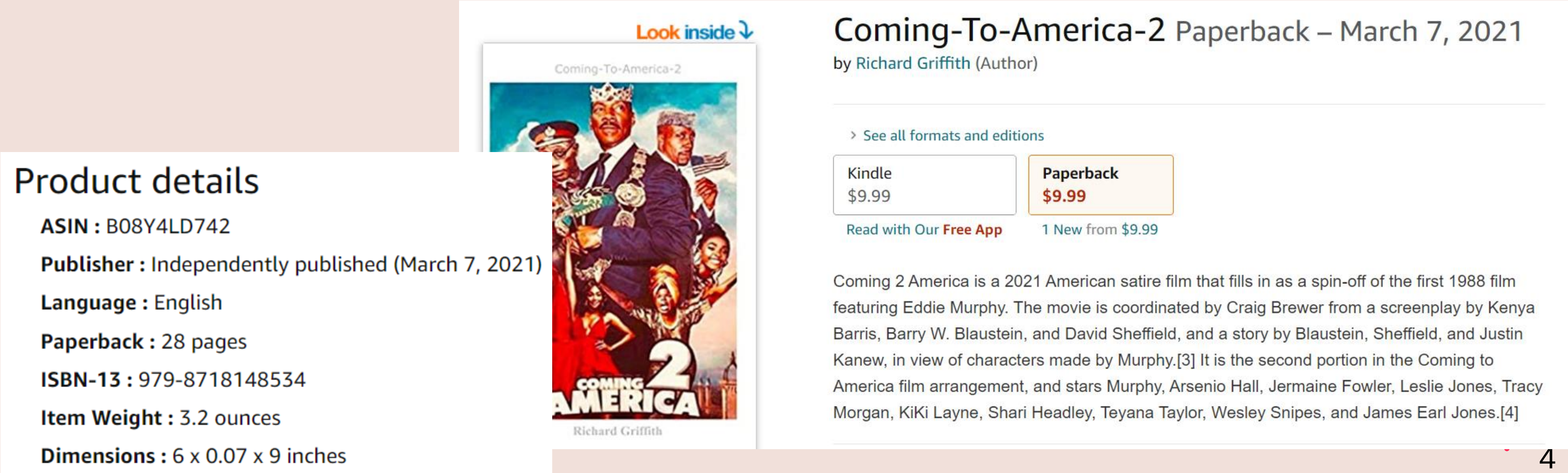
# Database THEORY

# What are Databases?

Holds a series of related information, reflected through tables

# Database Modeling Process: Requirements Gathering

- ○ Look at the system
- ○ Identify all of the details
- ○ Organize the details into categories
- ○ Identify information common to all categories
- ○ Identify information that is temporary



Coming-To-America-2 Paperback – March 7, 2021

by Richard Griffith (Author)

› See all formats and editions

| Kindle | Paperback |
|---|---|
| $9.99 | $9.99 |
| Read with Our Free App | 1 New from $9.99 |

Coming 2 America is a 2021 American satire film that fills in as a spin-off of the first 1988 film featuring Eddie Murphy. The movie is coordinated by Craig Brewer from a screenplay by Kenya Barris, Barry W. Blaustein, and David Sheffield, and a story by Blaustein, Sheffield, and Justin Kanew, in view of characters made by Murphy.[3] It is the second portion in the Coming to America film arrangement, and stars Murphy, Arsenio Hall, Jermaine Fowler, Leslie Jones, Tracy Morgan, KiKi Layne, Shari Headley, Teyana Taylor, Wesley Snipes, and James Earl Jones.[4]

Product details

**ASIN :** B08Y4LD742

**Publisher :** Independently published (March 7, 2021)

**Language :** English

**Paperback :** 28 pages

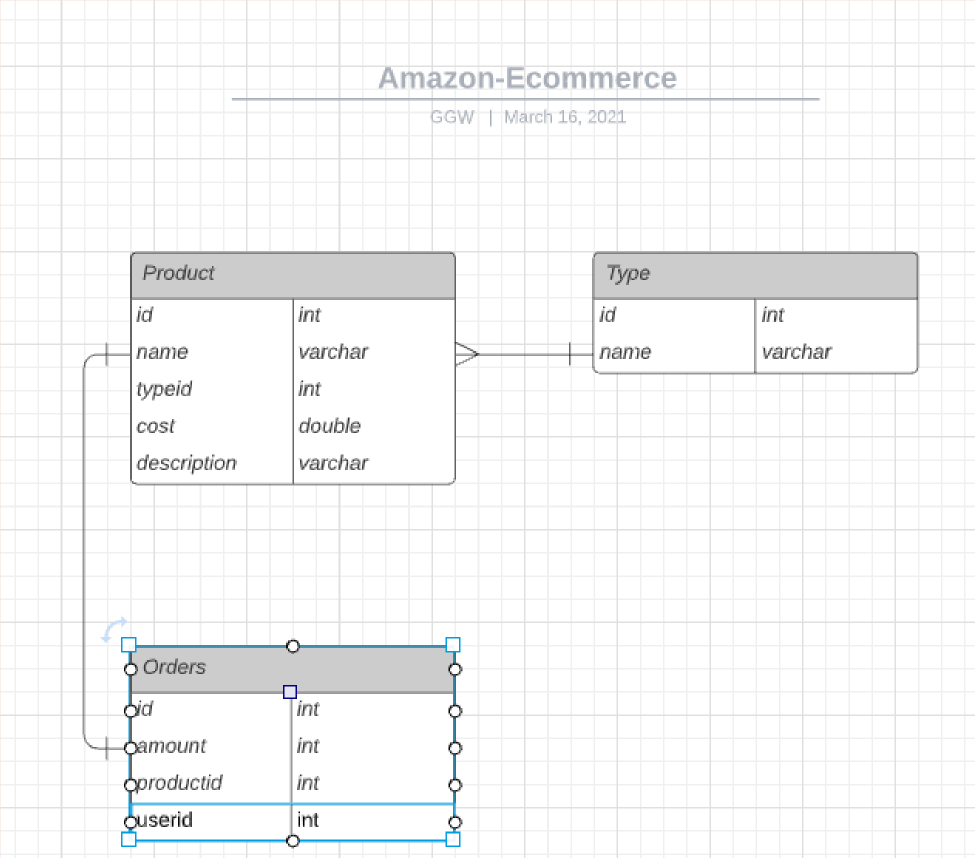
**ISBN-13 :** 979-8718148534

**Item Weight :** 3.2 ounces

**Dimensions :** 6 x 0.07 x 9 inches

# Database Modeling Process: Entity Design

- Create Entity Relationship Diagram
  - Software
    - Erwin
    - Visio
    - Toad Modeler
  - Online
    - LucidChart
    - Quick DB
    - SQL DBM
    - DB Diagram



Amazon-Ecommerce

GGW | March 16, 2021

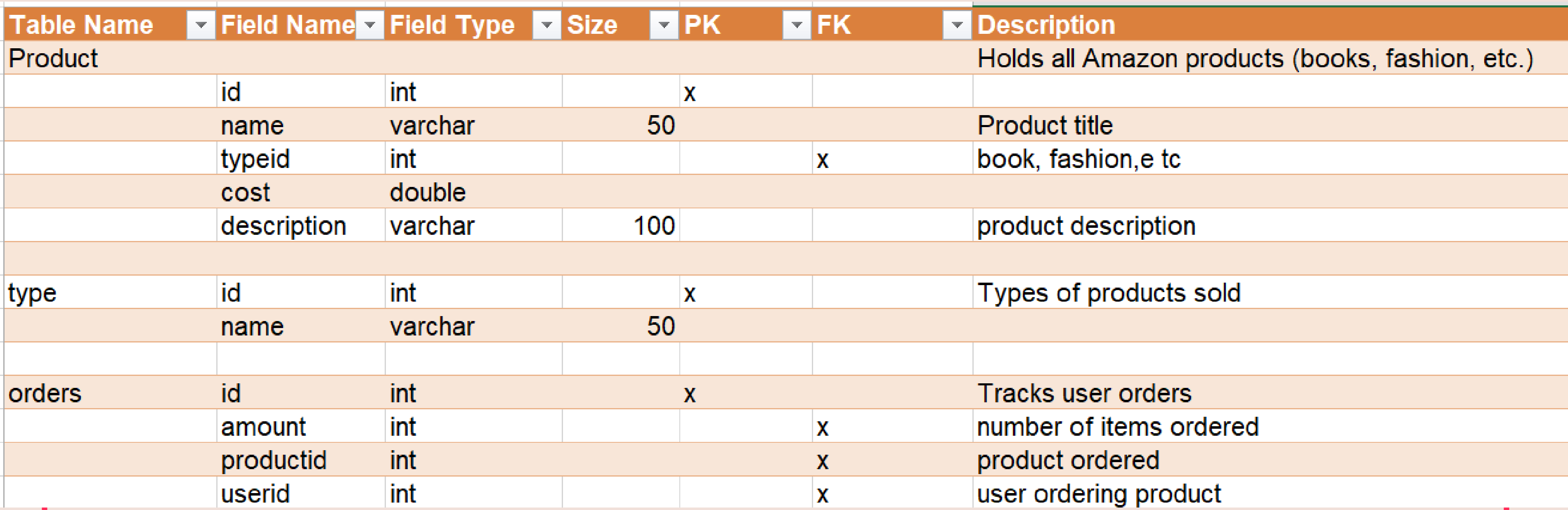
| Product | |
|---|---|
| id | int |
| name | varchar |
| typeid | int |
| cost | double |
| description | varchar |

| Type | |
|---|---|
| id | int |
| name | varchar |

| Orders | |
|---|---|
| id | int |
| amount | int |
| productid | int |
| userid | int |

# Database Modeling Process: Data Dictionary

Formally document the data structure



| Table Name | Field Name | Field Type | Size | PK | FK | Description |
|---|---|---|---|---|---|---|
| Product | | | | | | Holds all Amazon products (books, fashion, etc.) |
| | id | int | | x | | |
| | name | varchar | 50 | | | Product title |
| | typeid | int | | | x | book, fashion,e tc |
| | cost | double | | | | |
| | description | varchar | 100 | | | product description |
| | | | | | | |
| type | id | int | | x | | Types of products sold |
| | name | varchar | 50 | | | |
| | | | | | | |
| orders | id | int | | x | | Tracks user orders |
| | amount | int | | | x | number of items ordered |
| | productid | int | | | x | product ordered |
| | userid | int | | | x | user ordering product |

# Database Modeling: Data Schema

```
create table product (
 id integer   NOT NULL PRIMARY KEY,
 name  varchar(255),
 description varchar(255),
 cost     double,
 typeid  integer
 );

create table type (
 id integer   NOT NULL PRIMARY KEY,
   name  varchar(255));


create table user (
 id integer   NOT NULL PRIMARY KEY,
  username  varchar(255),
  lastname  varchar(255),
firstname  varchar(255),
email  varchar(255));
```

# Database: Field Types

| | |
|---|---|
| Words / paragraphs / names | Varchar |
| Numbers | Int / integer |
| Currency, Decimals | Float or Double |
| Images (binary image data) | Blobs |
| Large amounts of text (articles) | Large text |

# Vocabulary

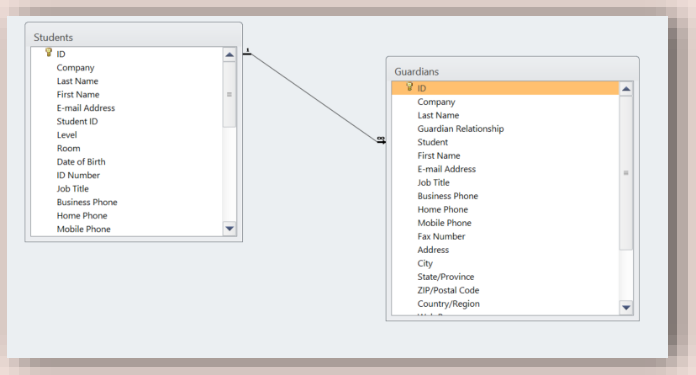
Each table might have ....

- Columns
    - Primary Key \ Composite Key - Unique, Required
    - Foreign Key - Points to other table
    - Data

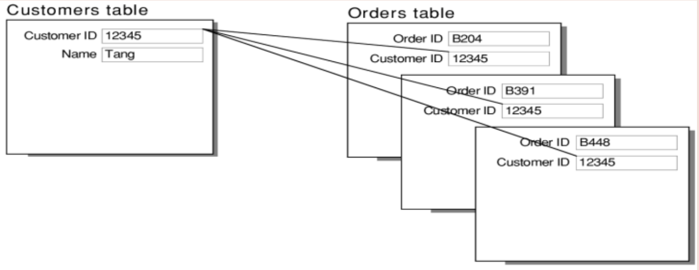
| **PRIMARY KEY** | | **FOREIGN KEY** |
|---|---|---|
| id (ID, INT) | Description (VARCHAR) | typeid (INT) |
| 1 | Chinese Shar-Pei | 1 |
| 2 | Italian Greyhound | 1 |
| 3 | Irish Water Spaniel | 2 |

# Relationship Types

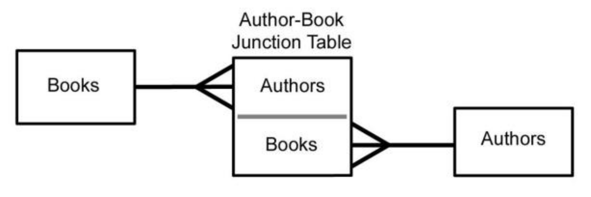
1-1



1-Many



Many-Many



# Database Download Break

- Download SQLite: https://sqlite.org/
- Create Database ECommerce
- Create Tables:
    - Product (productid, name, typeid, cost, description)
    - Type (typeid, name)
    - If time permits, Create User  (userid, lastname, firstname, username, email)

# Creating Tables

```
DROP TABLE IF EXISTS dogs;
CREATE TABLE dogs (
    dog_id integer PRIMARY KEY,
    breed varchar(50),
    type varchar(30),
    max_height integer,
    max_weight integer,
    max_life_span integer,
    general_health varchar(30),
    intelligence varchar(10),
    friendly varchar(10)
);
```

| Data Type | Description |
|---|---|
| INTEGER | Whole number between -2^31 and 2^31. |
| CHAR(size) | Fixed-length character string. Size is specified in parenthesis. Max 255 bytes. |
| VARCHAR(size) | Variable-length character string. Max size is specified in parenthesis and must be <= 65535. |
| DECIMAL(precision) | Number value with a max number of digits specified in parenthesis (must be <= 65). |
| DECIMAL(precision, scale) | Number value with a maximum number of digits of *precision*, with at most *scale* digits to the right of the decimal. |
| DATE | Date value |
| DATETIME | Date and time value |

# Inserting Data

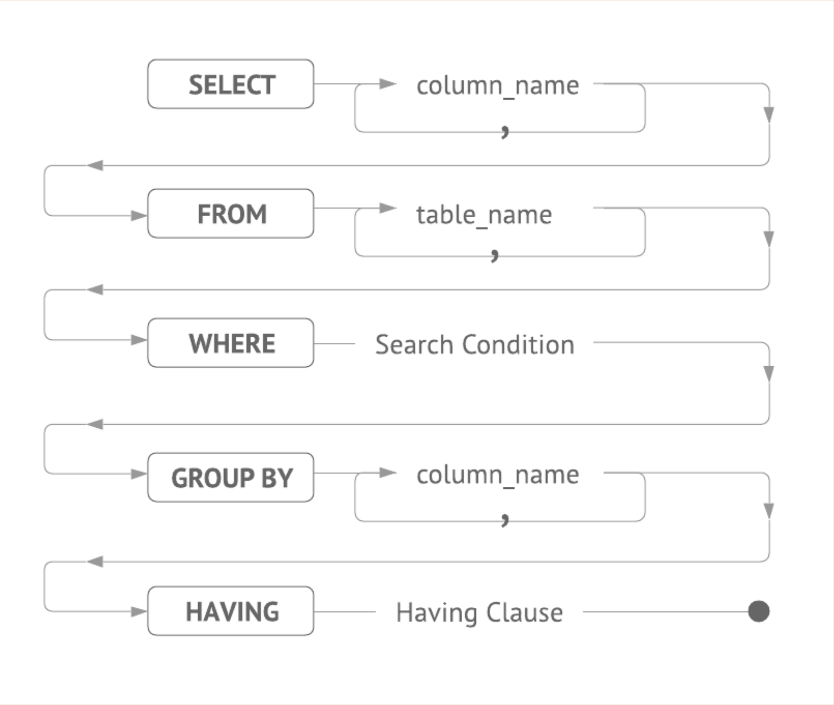
```
insert into type (id, name) values (1, 'book');
insert into type (id, name) values (2, 'dvd');

INSERT INTO product (id, name, typeid, cost, description)
VALUES
(1, 'Caged Bird', 1, 6.99, 'Self-novelization');

INSERT INTO product (id, name, typeid, cost, description)
VALUES
(2, 'Dude wheres my car', 2, 26.99, '90s comedy');
```

# Selecting Statement

Parts of SELECT statement

- **SELECT** clause: For Selection
- **FROM** clause: Which ‘Relation/Relations’ to use
- WHERE clause: For Join conditions, and Filtering

- ORDER BY: For sorting the output
- GROUP BY: For creating ‘data buckets’
- HAVING: For ‘filtering’ the ‘data buckets’



The simplest form of the SELECT statement contains the SELECT clause and the FROM clause.

# Selecting Data

```
SELECT * FROM table_name;

SELECT column1, column2, column3
FROM table_name;

SELECT column1, column2, column3
FROM table_name
WHERE conditional_selection;
```

# Sample Select Queries

**Simple queries**

```
SELECT dog_id, breed, type
FROM dogs LIMIT 5;
```

| dog_id | breed | type |
|---|---|---|
| 1 | Chinese Shar-Pei | Working Dogs |
| 2 | Italian Greyhound | Companion |
| 3 | Irish Water Spaniel | Sporting Dogs |
| 4 | Sussex Spaniel | Sporting Dogs |
| 5 | Affenpinscher | Companion |

```
SELECT dog_id, breed, max_weight
FROM dogs
WHERE max_weight > 175;
```

| dog_id | breed | max_weight |
|---|---|---|
| 84 | Irish Wolfhound | 180 |
| 181 | Great Dane | 200 |
| 187 | Mastiff | 220 |
| 188 | Neapolitan Mastiff | 200 |
| 193 | Saint Bernard | 180 |

```
SELECT dog_id, breed, type
FROM dogs
WHERE breed LIKE '%German% ';
```

| dog_id | breed | type |
|---|---|---|
| 57 | German Shepherd | Herding Dogs |
| 124 | German Shorthaired Pointer | Sporting Dogs |
| 125 | German Wirehaired Pointer | Sporting Dogs |
| 179 | German Pinscher | Working Dogs |

# More Select Queries

**Simple queries**

```
select * from products where cost is not null
select * from products where cost is null
select * from products where typeid != 1

select * from products where cost > 5 order by cost
select * from products where cost > 5 order by cost, name
select * from products where name like '%angelou%'
select *, count(*) from products  group by typeid
```