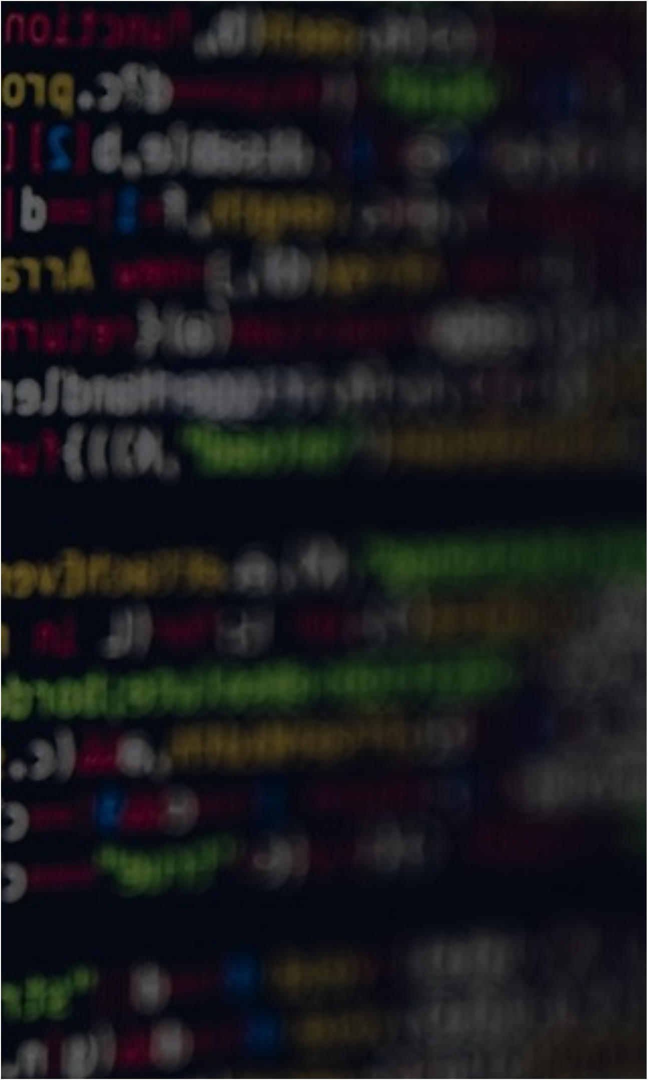
EXPLORE DIGITAL SKILLS

SQL for Data Science
Predict



Contents

Problem Context/Domain

Predict Rules + Instructions

Student Starter Pack

Important Packages

Data ERD Guide

Predict FAQs

EXPLORE DIGITAL SKILLS

Problem Context/Domain - Retail (Online Retailing Business)

Problem Statement:

The Bhejane Trading store is an online retailer specializing in the sale of covid-related essential items. As a consultant hired by the company, you have been tasked with the objective of normalizing the database of the store’s inventory management system.

You are provided with an unnormalised database, and are expected to normalise it's contents to bring it into 3rd Normal Form (**3NF**). The database has 2 tables (products and transactions) which are summarised here.

Deliverables:

After having normalised the DB, you will be required to answer several multiple-choice questions which test your completed work, and your practical SQL skills gained in the course.

NB: The following deliverables must also be **uploaded to Athena**.

- A notebook containing SQL queries that transform the database from an unnormalised form to **3NF.**
- The notebook should also contain queries used to answer the MCQ.
- A SQLite ‘.db’ database file of the normalised database.

Predict Rules + Instructions

- This project is an **individual** project; your work needs to reflect your understanding of the course content.

- You are free to share ideas with colleagues and classmates, however, **you are not allowed to share your code, solutions, or submissions with any other individual**. Plagiarism will not be tolerated.

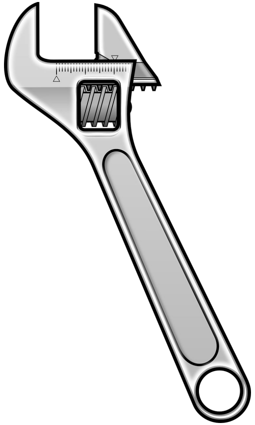
- You are required to **submit all of the code** you use to both normalise the given dataset, and to answer the related MCQ assessment. Submit your completed *starter notebook*, along with any other material as a **zipped file** under the 'Upload Predict File' tab on Athena.

- **The official due date of the Predict will be displayed on Athena**. No submissions after 23:59 on this date will be accepted for marking.

Student Starter Pack - Getting You on the Right Track

In order to help you get your bearings within the Predict, we've prepared a 'starter pack' which contains essential material to guide your work. This material includes:

- **Base notebook:** A Jupyter notebook containing code and instructions to begin work on the Predict. Continue developing this file to use for final solution submission to Athena.
- **The unnormalised data:** Two .csv files containing the unnormalised data.
  - 'bhejane_covid_essentials_Products.csv'
  - 'bhejane_covid_essentials_Transactions.csv'
- **A description of the various data fields found in the database.**

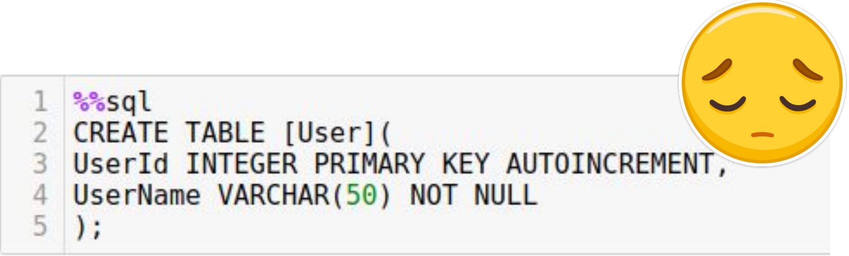


EXPLORE | DIGITAL SKILLS

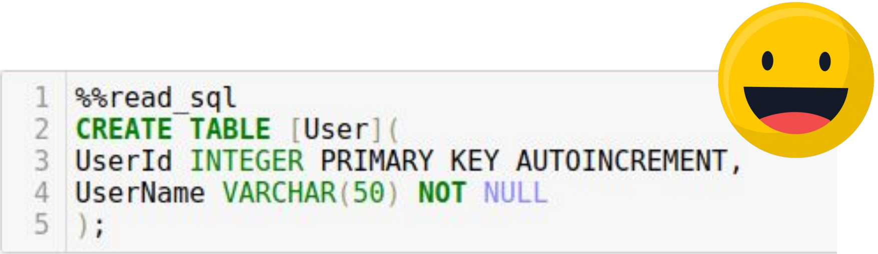
Making Sense of our Queries

Within this Predict we'll be writing a *lot* of SQL statements. In order to make your SQL queries more human-readable and to help you along, we will install a **sql_magic** package to assist with syntax highlighting.

- Install the sql_magic package by entering the following command into your terminal: `pip install sql_magic`.
- Now you can use the ***%%read_sql*** magic command at the start of each cell when writing your SQL queries and the syntax will be highlighted.



```
%%sql
CREATE TABLE [User](
UserId INTEGER PRIMARY KEY AUTOINCREMENT,
UserName VARCHAR(50) NOT NULL
);
```

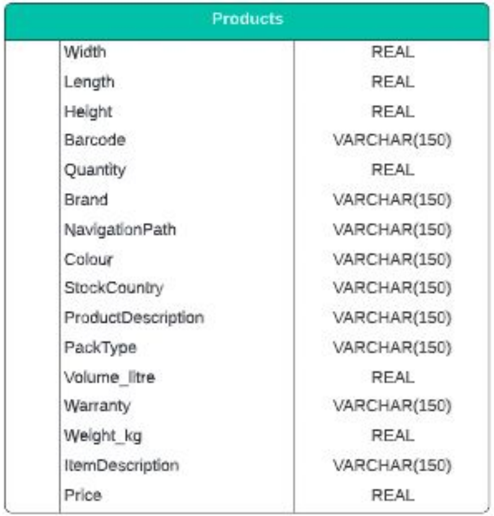


```
%%read_sql
CREATE TABLE [User](
UserId INTEGER PRIMARY KEY AUTOINCREMENT,
UserName VARCHAR(50) NOT NULL
);
```

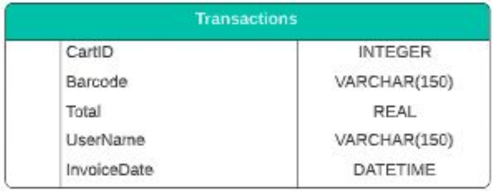
Detailing the Data - Original Database Tables

To help familiarise yourself with the data in the original database, we provide the following ERD - showing the various fields for the **Products** and **Transactions** tables respectively.

You are required to use the principles of database normalisation to transform these tables into the **3NF** schema. Subsequent slides will detail the normalization process

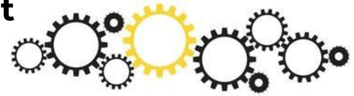


Products	
Width	REAL
Length	REAL
Height	REAL
Barcode	VARCHAR(150)
Quantity	REAL
Brand	VARCHAR(150)
NavigationPath	VARCHAR(150)
Colour	VARCHAR(150)
StockCountry	VARCHAR(150)
ProductDescription	VARCHAR(150)
PackType	VARCHAR(150)
Volume_litre	REAL
Warranty	VARCHAR(150)
Weight_kg	REAL
ItemDescription	VARCHAR(150)
Price	REAL



Transactions	
CartID	INTEGER
Barcode	VARCHAR(150)
Total	REAL
UserName	VARCHAR(150)
InvoiceDate	DATETIME

***NB: Be wary of handling NULL values in the dataset**

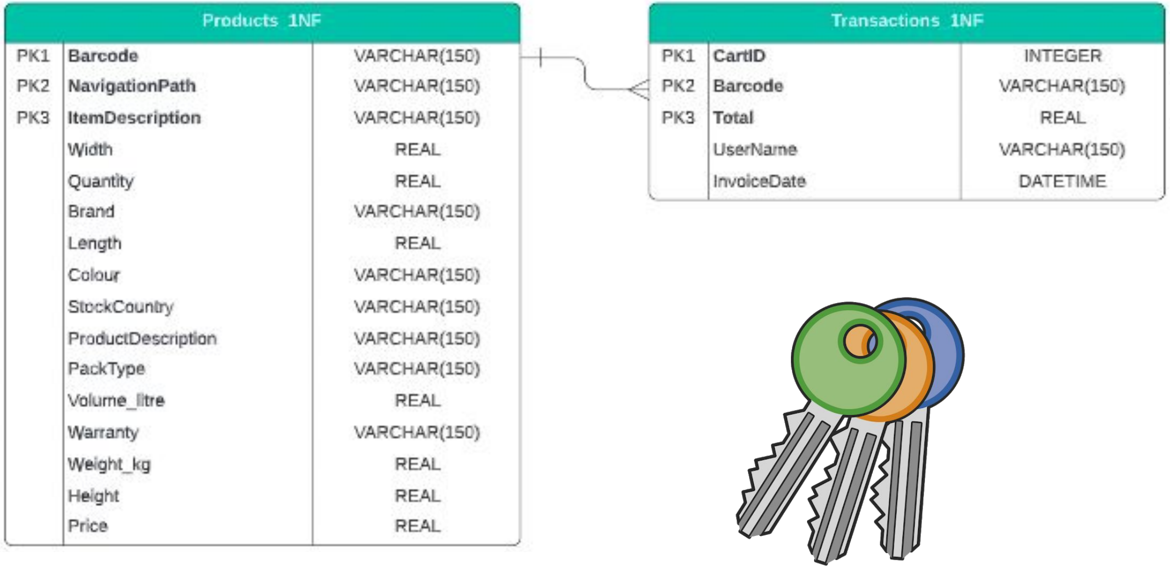


Detailing the Data - 1NF Entity Relationship Diagram

Throughout the Predict you will be given the target ERD for each normalization step.

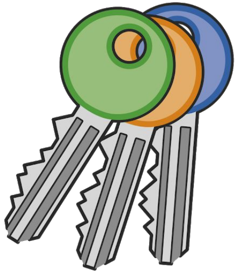
To the right is the ERD sketch for the 1st Normal Form to get you started.

Pay attention to field attributes such as **data types, primary keys, composite keys, foreign keys** and **relationships** that exist amongst them

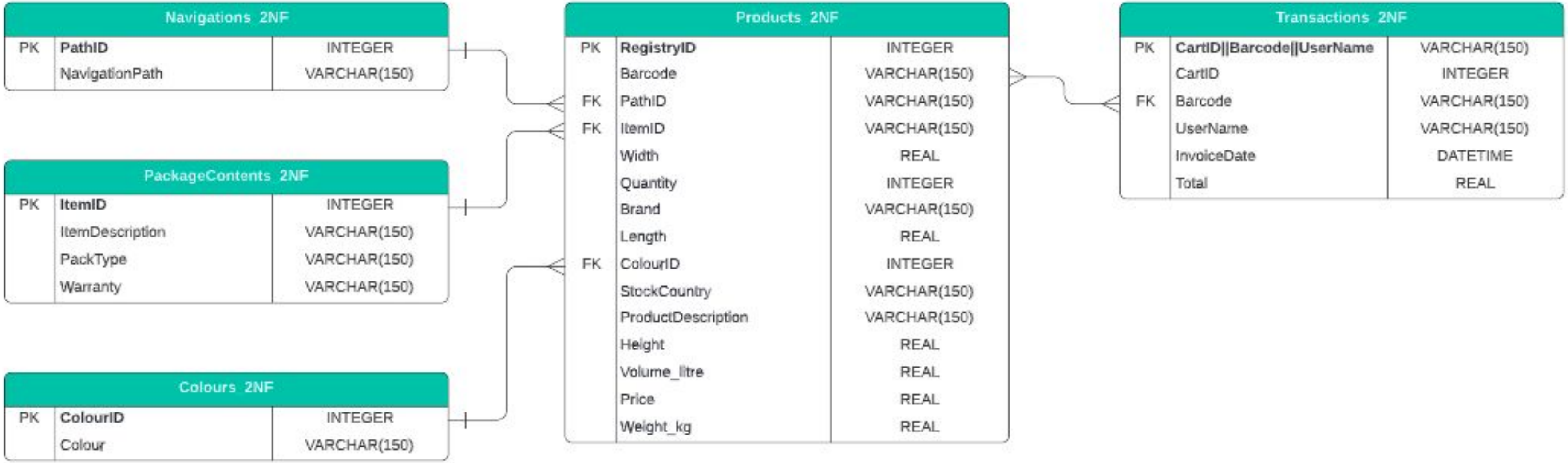


Products 1NF		
PK1	**Barcode**	VARCHAR(150)
PK2	**NavigationPath**	VARCHAR(150)
PK3	**ItemDescription**	VARCHAR(150)
	Width	REAL
	Quantity	REAL
	Brand	VARCHAR(150)
	Length	REAL
	Colour	VARCHAR(150)
	StockCountry	VARCHAR(150)
	ProductDescription	VARCHAR(150)
	PackType	VARCHAR(150)
	Volume_litre	REAL
	Warranty	VARCHAR(150)
	Weight_kg	REAL
	Height	REAL
	Price	REAL

Transactions 1NF		
PK1	**CartID**	INTEGER
PK2	**Barcode**	VARCHAR(150)
PK3	**Total**	REAL
	UserName	VARCHAR(150)
	InvoiceDate	DATETIME



Detailing the Data - 2NF Entity Relationship Diagram



Navigations_2NF		
PK	**PathID**	INTEGER
	NavigationPath	VARCHAR(150)

PackageContents_2NF		
PK	**ItemID**	INTEGER
	ItemDescription	VARCHAR(150)
	PackType	VARCHAR(150)
	Warranty	VARCHAR(150)

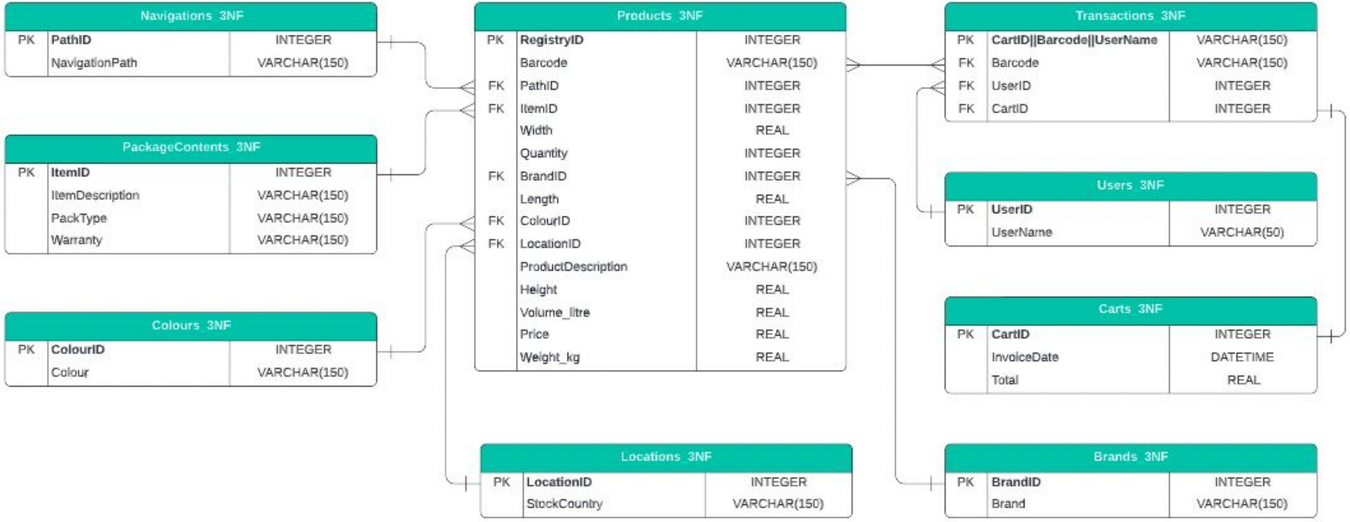
Colours_2NF		
PK	**ColourID**	INTEGER
	Colour	VARCHAR(150)

Products_2NF		
PK	**RegistryID**	INTEGER
	Barcode	VARCHAR(150)
FK	PathID	VARCHAR(150)
FK	ItemID	VARCHAR(150)
	Width	REAL
	Quantity	INTEGER
	Brand	VARCHAR(150)
	Length	REAL
FK	ColourID	INTEGER
	StockCountry	VARCHAR(150)
	ProductDescription	VARCHAR(150)
	Height	REAL
	Volume_litre	REAL
	Price	REAL
	Weight_kg	REAL

Transactions_2NF		
PK	**CartID||Barcode||UserName**	VARCHAR(150)
	CartID	INTEGER
FK	Barcode	VARCHAR(150)
	UserName	VARCHAR(150)
	InvoiceDate	DATETIME
	Total	REAL

You are encouraged to use the **AUTOINCREMENT** property when creating new fields that are going to be used as primary keys.

Detailing the Data - 3NF Entity Relationship Diagram



Navigations 3NF

PK	**PathID**	INTEGER
	NavigationPath	VARCHAR(150)

PackageContents 3NF

PK	**ItemID**	INTEGER
	ItemDescription	VARCHAR(150)
	PackType	VARCHAR(150)
	Warranty	VARCHAR(150)

Colours 3NF

PK	**ColourID**	INTEGER
	Colour	VARCHAR(150)

Products 3NF

PK	**RegistryID**	INTEGER
	Barcode	VARCHAR(150)
FK	PathID	INTEGER
FK	ItemID	INTEGER
	Width	REAL
	Quantity	INTEGER
FK	BrandID	INTEGER
	Length	REAL
FK	ColourID	INTEGER
FK	LocationID	INTEGER
	ProductDescription	VARCHAR(150)
	Height	REAL
	Volume_litre	REAL
	Price	REAL
	Weight_kg	REAL

Locations 3NF

PK	**LocationID**	INTEGER
	StockCountry	VARCHAR(150)

Transactions 3NF

PK	**CartID||Barcode||UserName**	VARCHAR(150)
FK	Barcode	VARCHAR(150)
FK	UserID	INTEGER
FK	CartID	INTEGER

Users 3NF

PK	**UserID**	INTEGER
	UserName	VARCHAR(50)

Carts 3NF

PK	**CartID**	INTEGER
	InvoiceDate	DATETIME
	Total	REAL

Brands 3NF

PK	**BrandID**	INTEGER
	Brand	VARCHAR(150)

Hint:

As you progress through the different normal forms you may find it easier populate the current normal forms using the previous normal forms

EXPLORE DIGITAL SKILLS

Predict-related FAQs

This page will be updated periodically with common predict-related questions which may arise during the Sprint. Consider consulting this space before asking your course facilitator a question.

Considerations to keep in mind when completing the predict, before answering the predict questions.

1. The aim of the predict is to understand and implement normalization on the dataset provided. This includes, understanding separation of entities (tables which serve a single purpose), maintaining relationships and enforcing normalization through data integrity.

2. Following the normalization process is an important step to follow in order to be able to answer the predict questions effectively.

3. Having an understanding of your problem and data can be very helpful in guiding your thinking to solve a problem. At each stage of the normalization it is suggested that you take some time to reflect on what changes were made from the previous normal form and understand why transformations were made.

Predict-related FAQs

I am getting the following error - *ModuleNotFoundError: No module named 'sql_magic';* What should I do?

- Please make sure that you have installed the **sql_magic** using the following command: ***pip install sql_magic***

I cannot make changes to my table creation code, I get the following error everytime I try - *OperationalError: table <TableName> already exists*

- You are advised to first drop the old table before re-creating the table with your new changes
  - **DROP TABLE IF EXISTS** [TableName];

- You can drop and create the tables as many time as you want, just remember to keep the table naming convention consistent with the ERD sketches that are provided.

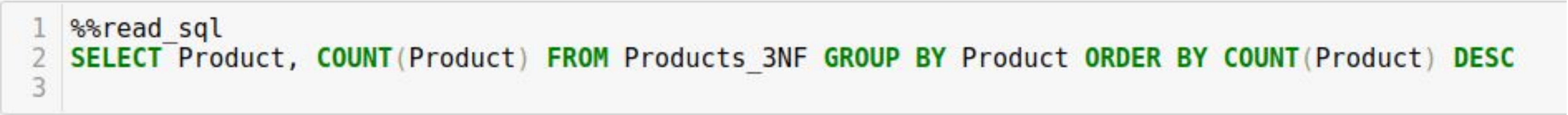
What does ‘PK’ and ‘FK’ stand for when looking at the ERD sketches?

- **PK:** Primary Key
- **FK:** Foreign Key

Predict-related FAQs

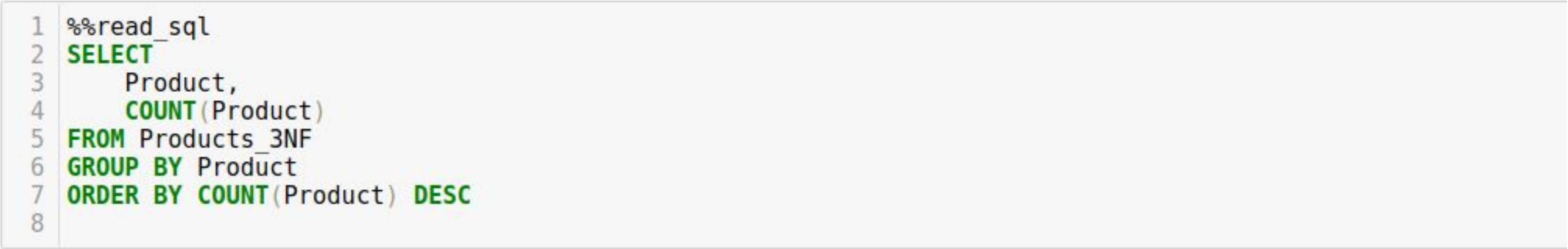
I am constantly getting errors and debugging is a nightmare

- SQL by nature requires one to be pedantic - so pay special attention to syntax and formatting. If your SQL queries generally look like the below - *may the debugging gods be with you…*



```
%%read_sql
SELECT Product, COUNT(Product) FROM Products_3NF GROUP BY Product ORDER BY COUNT(Product) DESC

```

- SQL doesn’t have any formatting rules (such as indentation in python), so it will allow you to run the above query with no issues at all. It is however recommended you practise good SQL hygiene and stay away from this practice. Although there is no book of all truths for SQL formatting, it should generally take the following form:



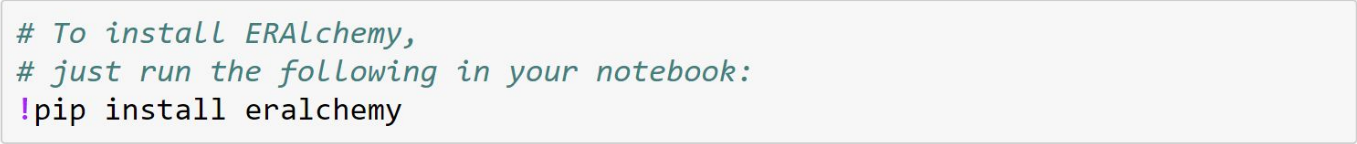
```
%%read_sql
SELECT
    Product,
    COUNT(Product)
FROM Products_3NF
GROUP BY Product
ORDER BY COUNT(Product) DESC

```

Predict-related FAQs

How can I compare my normalised database to the reference ERD diagrams?

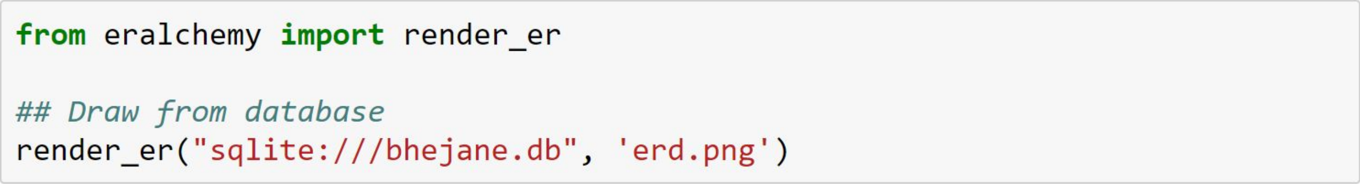
- ERAlchemy is a useful package for viewing relationship diagrams within Jupyter

  ERAlchemy requires GraphViz to generate the graphs and Python. Both are available for Windows, Mac and Linux.



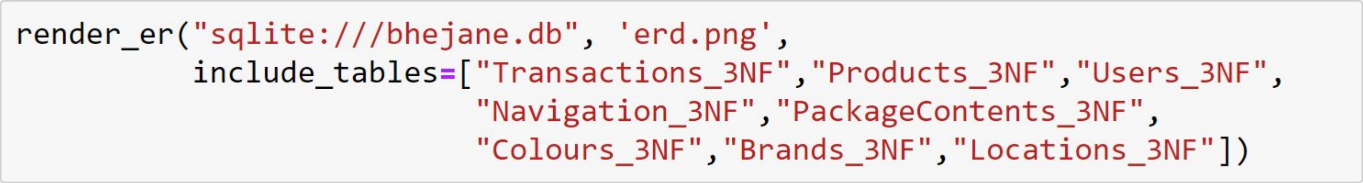
```
# To install ERAlchemy,
# just run the following in your notebook:
!pip install eralchemy
```

- Within a Jupyter codecell, execute the render_er Python function to see your relationship diagram



```
from eralchemy import render_er

## Draw from database
render_er("sqlite:///bhejane.db", 'erd.png')
```

- Or be more specific on the tables you want to include in the output



```
render_er("sqlite:///bhejane.db", 'erd.png',
          include_tables=["Transactions_3NF","Products_3NF","Users_3NF",
                          "Navigation_3NF","PackageContents_3NF",
                          "Colours_3NF","Brands_3NF","Locations_3NF"])
```